Università degli studi di Padova
Dipartimento di Scienze Statistiche
Corso di Laurea Triennale in

Statistica, Economia e Finanza

Relazione Finale

Trasformazione Box-Cox: un'analisi basata sulla verosimiglianza

Relatore Prof. Nicola Sartori
Dipartimento di Scienze Statistiche

Laureando Tommaso Rigon
Matricola N 1010510

Anno Accademico 2012/2013

Indice

Elenco dei codici

Elenco delle tabelle

Introduzione

Uno degli strumenti più utilizzati in statistica è il modello lineare. La sua utilità e diffusione giustifica gli sforzi atti a comprendere il suo funzionamento. In questa relazione, di carattere prevalentemente teorico, viene analizzata una particolare tecnica utilizzata nei modelli lineari: la trasformazione Box-Cox. Questo approfondimento si è reso necessario poichè, nell'uso quotidiano, essa viene utilizzata trascurando volutamente alcuni aspetti. Lo scopo è mostrare come questo approccio porti a distorsioni rilevanti. Pur essendo già noti alcuni risultati nella letteratura statistica, questi vengono ripresi e illustrati dettagliatamente, insieme ad alcuni esempi. Contestualmente, si forniscono dei codici che consentono di ripercorrere i calcoli effettuati e condurre analisi su dataset differenti.

Gli argomenti riguardanti la verosimiglianza compaiono trasversalmente e proprio per questo nel primo capitolo si presenta una rassegna delle principali nozioni connesse ad essa. Non sono informazioni direttamente collegate alla relazione ma strettamente necessarie per la sua comprensione. Viene inoltre definita la notazione delle principali quantità di verosimiglianza.

Nel secondo capitolo vengono brevemente presentati il modello lineare e la trasformazione Box-Cox. Si tratta di un' introduzione volta a esporre i motivi per cui, in certi casi, è necessario utilizzare una trasformazione di variabile tra quelle note. Quella indicata da Box-Cox occupa una posizione di rilievo e viene infatti utilizzata, nel modo consueto, in un dataset, a scopo esemplificativo.

Il terzo capitolo è il nocciolo della relazione. Si affrontano i calcoli analitici che permettono di raggiungere le quantità di verosimiglianza specifiche della trasformazione Box-Cox. Si discute poi la sua legittimità che porterà al suo

parziale abbandono, seguendo le argomentazioni di Bickel e Docksum.

Nel quarto capitolo infine si segnalano, da un punto di vista empirico, le distorsioni che l'approccio usuale comporta, riprendendo il dataset usato nel secondo capitolo. Per far ciò viene prima controllato, tramite simulazioni, il codice che ha reso possibile il calcolo di queste distorsioni.

Nel quinto capitolo si conclude riassumendo i risultati a cui si è giunti.

Capitolo 1

L'inferenza di verosimiglianza

1.1 Introduzione

Fare inferenza significa, letteralmente, trarre delle conclusioni sulla base di alcune verità. In ambito statistico il punto di partenza sono i dati, da cui si cerca di estrarre qualche informazione di carattere generale. Le quantità d'interesse sono molteplici e differiscono a seconda del contesto applicativo. Ad esempio, spesso si è interessati a studiare l'andamento medio di un fenomeno. Dato che il punto di partenza sono indicazioni parziali, o meglio, **quantità campionarie**, è naturale che le conclusioni che se ne trarranno saranno soggette ad errore. V'è quindi la necessità di una tecnica che permetta di ottenere stime il più precise possibile e di cui si sia in grado di determinare l'affidabilità.

L'inferenza basata sulla verosimiglianza è uno strumento estremamente versatile, introdotto da Ronald Fisher (1890-1962), che presume l'esistenza di un modello statistico $\mathcal{F}$. In questo capitolo introduttivo verranno presentate alcune nozioni basilari e definite le notazioni per le quantità utilizzate.

La presentazione è basata sui testi Azzalini (2008, cap. 2-4) e Pace e Salvan (2001, cap. 1-6).

1.2 Specificazione del modello

La prima fondamentale assunzione è considerare il vettore delle osservazioni $y = (y_1, \ldots, y_n)$ come una realizzazione di una variabile casuale Y con distribuzione di probabilità $p_0(y)$. Il modello generatore dei dati è ignoto ma spesso si possono esprimere delle considerazioni sul fenomeno in esame. Data la natura stocastica di Y, la stima del modello generatore dei dati è tanto più accurata quanto più si riesce a restringere la classe $\mathcal{F}$ alla quale $p_0(y)$ appartiene. Il modello si intende correttamente specificato se $p_0(y) \in \mathcal{F}$.

Il modello $\mathcal{F}$ può appartenere a una delle seguenti classi: *modello non parametrico, modello semi-parametrico, modello parametrico*. Nel seguito ci si occuperà unicamente di quest'ultimo, la cui definizione formale è:

$$\mathcal{F} = \{p(y, \theta) : \theta \in \Theta \subseteq \mathbb{R}^p\},$$

per qualche $p \in \mathbb{N}$. Lo spazio Θ è chiamato **spazio parametrico**. Si deve ipotizzare che esista una relazione biunivoca tra ciascun modello ed il valore assunto dal parametro. Questa proprietà è chiamata **identificabilità**.

1.3 Assuzioni e notazioni

Se $\mathcal{F}$ è un modello parametrico per i dati y con funzione del modello $p_y(y; \theta)$ e $\theta = (\theta_1, \ldots, \theta_p) \in \Theta \subseteq \mathbb{R}^p$, la **funzione di verosimiglianza** $L : \Theta \to \mathbb{R}^+$ è definita come:

$$L(\theta; y) = c(y)p(y; \theta),$$

dove $c(y)$ è una costante che non dipende da θ. La funzione $L(\theta; y)$ ha quindi la stessa forma della funzione di probabilità del modello in cui però y è fissato e θ è libero di variare. Pur assumendo valori che corrispondono a probabilità, $L(y; \theta)$ non è quindi una funzione di probabilità.

La verosimiglianza è in realtà una classe di funzioni equivalenti che differiscono per una costante moltiplicativa $c(y)$ e si può parlare dunque di **verosimiglianze equivalenti**.

Il sostegno empirico a favore di θ_1 rispetto a θ_2 è misurato dal rapporto

$$\frac{L(\theta_1; y)}{L(\theta_2; y)},$$

che è chiamato **rapporto di verosimiglianza**.

L'interpretazione di $L(\theta; y)$ è la seguente: a partire dalle osservazioni y è possibile stabilire quale tra θ_1 e θ_2 è più verosimile. Segue quindi che θ_1 è preferibile a θ_1 se $L(\theta_1; y) \geq L(\theta_2; y)$. Ad esempio nella Figura 1.1, $\theta = 1.9$ è preferibile a $\theta = 1.2$, infatti: $L(1.9; y) \geq L(1.2; y)$.

Un'ipotesi frequente è che i dati $y = (y_1 \ldots, y_n)$ siano realizzazioni *i.i.d*, ovvero osservazioni indipendenti ed identicamente distribuite. Perciò se $p(y_i; \theta)$ è la distribuzione di probabilità marginale per la singola osservazione, la funzione di verosimiglianza diventa

$$L(\theta; y) = \prod_{i=1}^{n} p(y_i; \theta).$$

Funzione di log-verosimiglianza

La funzione di log-verosimiglianza viene introdotta perchè permette maggiore facilità di calcolo pur mantenendo tutta l'informazione contenuta in $L(\theta; y)$. È definita come

$$l(\theta; y) = \log L(\theta; y),$$

dove $\log(\cdot)$ è il logaritmo naturale. Se ne ricava inoltre la quantità

$$l(\theta; y) = c'(y) + \log p(y; \theta),$$

da cui si deduce che le log-verosimiglianze si equivalgono a meno di una costante additiva, costituendo anch'esse una classe di funzioni equivalenti. Nel caso il campione sia costituito da osservazioni *i.i.d.*, per le proprietà dei logaritmi ne consegue che

$$l(\theta; y) = \sum_{i=1}^{n} \log p(y_i; \theta).$$

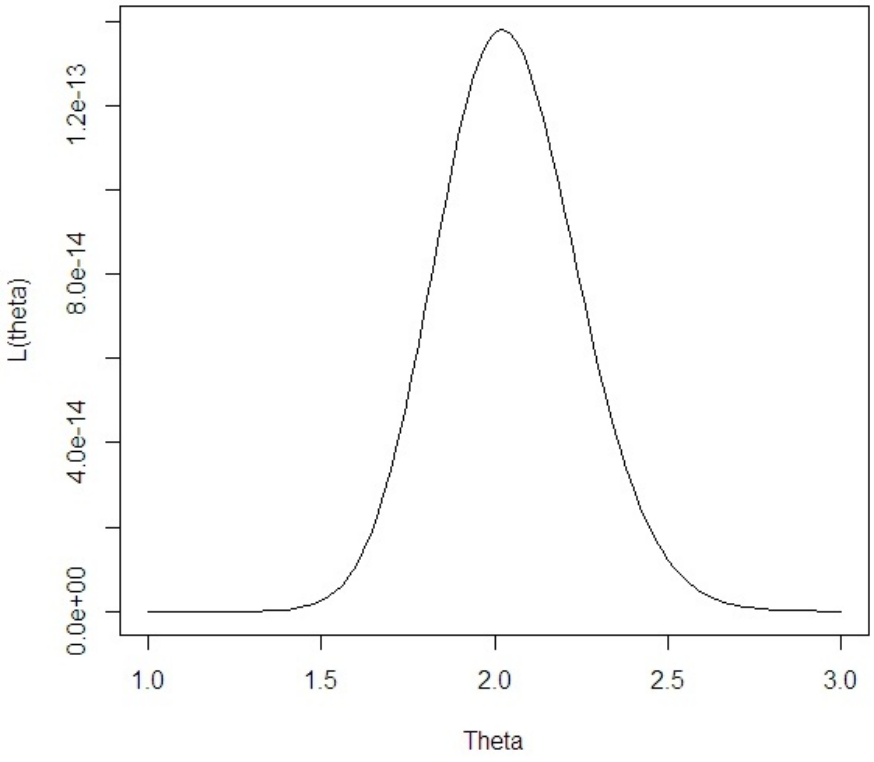


Figura 1.1: Funzione di verosimiglianza esponenziale di media 0.5 con $n = 50$

1.4 Quantità di verosimiglianza

1.4.1 Stima di massima verosimiglianza

Seguendo le condizioni poste in Azzalini (2008, pag. 88), si è di fronte ad un problema regolare di stima se:

- il modello è identificabile;
- lo spazio parametrico Θ è un intervallo aperto di $\mathbb{R}^p$;
- le funzioni di probabilità (o densità) hanno tutte lo stesso supporto;
- per la funzione di densità f si può scambiare due volte il segno di integrale con quello di derivata rispetto a θ.

In realtà queste sono condizioni piuttosto deboli che si verificano nella gran parte dei casi pratici. Nel caso oggetto di questa tesi, le condizioni di regolarità sono soddisfatte.

Il valore $\hat{\theta}$ è detto **stima di massima verosimiglianza** se

$$L(\hat{\theta}) \geq L(\theta), \quad \forall \theta \in \Theta,$$

ovvero se $\hat{\theta}$ è il punto di massimo assoluto per la funzione di verosimiglianza. Poiché $\log(\cdot)$ è una trasformazione monotona crescente, ne consegue che $\hat{\theta}$ è punto di massimo per $L(\theta)$ se e solo se lo è per $l(\theta)$.

1.4.2 Funzione punteggio

Si definisce funzione score o punteggio

$$\frac{\partial l(\theta)}{\partial \theta} = l_*(\theta). \tag{1.1}$$

Sotto le condizioni di regolarità il punto di massima verosimiglianza va cercato tra le soluzioni dell'equazione $l_*(\theta) = 0$, che prende il nome di **equazione di verosimiglianza**. Inoltre, vale che

$$\mathbf{E}[l_*(\theta)] = 0 \qquad \text{e che} \qquad \mathrm{Var}[l_*(\theta)] = \mathbf{E}[l_*(\theta) l_*(\theta)^T] = I(\theta).$$

1.4.3 Informazione osservata e informazione attesa

La matrice di informazione osservata $j(\theta)$ e quella di informazione attesa $I(\theta)$ di dimensioni $p \times p$ sono definite nel modo seguente

$$j(\theta) = -l_{**}(\theta) = \frac{\partial l(\theta)}{\partial \theta^T \partial \theta}, \qquad I(\theta) = \mathbf{E}[j(\theta)]. \tag{1.2}$$

Per indicare la matrice inversa $I(\theta)^{-1}$ nel blocco (τ, τ) si è soliti indicarla come $I(\theta)^{\tau\tau}$.

1.5 Alcuni risultati asintotici

Fino ad ora non si è detto nulla circa i vantaggi di utilizzare un approccio di verosimiglianza rispetto ad altri metodi. È auspicabile che lo stimatore fornisca una stima sempre più precisa dei parametri e che converga al vero valore al divergere della numerosità campionaria. Sotto le condizioni di regolarità è possibile dimostrare alcune proprietà dello stimatore di massima verosimiglianza. In particolare, $\hat{\theta}(Y)$ è:

- asintoticamente non distorto, ovvero: $\lim_{n\to\infty} \mathbf{E}_\theta[\hat{\theta}(Y)] = \theta$. In campioni finiti è, in generale, distorto;
- consistente, ovvero converge in probabilità a θ, $\hat{\theta}(Y) \xrightarrow{p} \theta$ e perciò: $\lim_{n\to\infty} \mathrm{Var}(\hat{\theta}(Y)) = 0$;
- asintoticamente efficiente.

Qualora non fosse possibile risalire alla distribuzione esatta dello stimatore, è possibile ricorrere ad alcuni risultati asintotici. Per n sufficientemente grande vale che

$$\hat{\theta} \dot{\sim} N_p(\theta, I(\theta)^{-1}). \tag{1.3}$$

La matrice $I(\theta)$ in generale non è nota e quindi la si può sostituire sia con $I(\hat{\theta})$ che con $j(\hat{\theta})$ che ne costituiscono delle stime consistenti.

Questo risultato è di fondamentale importanza perché ci si riconduce, almeno approssimativamente, ad una distribuzione nota. Ciò permette di costruire test d'ipotesi ed intervalli di confidenza anche quando la distribuzione esatta è troppo complessa da ricavare analiticamente.

1.5.1 Test collegati alla verosimiglianza

In un sistema d'ipotesi in cui si ha che $H_0 : \theta = \theta_0$ e $H_1 : \theta \neq \theta_0$ esistono tre test statistici asintoticamente equivalenti che, sotto H_0, al divergere di n, convergono in distribuzione ad una distribuzione χ^2_p. Il primo è chiamato **log-rapporto di verosimiglianza** ed è pari a

$$W(\theta_0) = 2(l(\hat{\theta}) - l(\theta_0)) = -2(l(\theta_0) - l(\hat{\theta})). \tag{1.4}$$

Il secondo è chiamato **test di Wald** ed è pari a

$$W_e(\theta_0) = (\hat{\theta} - \theta_0)^T I(\theta)(\hat{\theta} - \theta_0), \tag{1.5}$$

inoltre il terzo è chiamato **test score** o **test di Rao** ed è pari a

$$W_u(\theta_0) = l_*(\theta_0) I(\theta_0)^{-1} l_*(\theta_0). \tag{1.6}$$

Nel caso in cui $p = 1$ si ricorre alle versioni unilaterali di questi stessi test $r(\theta_0), r_e(\theta_0), r_u(\theta_0)$, che, sotto H_0, si distribuiscono approssimativamente come delle $N(0, 1)$. Le definizioni dei test appena citati sono rispettivamente

$$\begin{aligned} r(\theta_0) &= \operatorname{sgn}(\hat{\theta} - \theta_0)\sqrt{W(\theta_0)}, \\ r_e(\theta_0) &= (\hat{\theta} - \theta_0)\sqrt{I(\hat{\theta})}, \\ r_u(\theta_0) &= \operatorname{sgn}(\hat{\theta} - \theta_0) l_*(\theta_0) I(\theta_0)^{-\frac{1}{2}}. \end{aligned}$$

1.6 Verosimiglianza profilo

In alcuni casi pratici, in cui $p > 1$, l'attenzione è rivolta verso un primo gruppo di parametri detti *d'interesse*, indicati con τ. Tuttavia si deve comunque tener conto dell'esistenza dei parametri del secondo gruppo, detti *di disturbo*, indicati con ζ.

Se $\zeta = \zeta_0$, ovvero se fosse noto, la funzione di verosimiglianza sarebbe $L(\tau, \zeta_0)$. Ciò tipicamente non avviene e v'è quindi la necessità di introdurre una sorta di surrogato per la funzione di verosimiglianza propria in cui ζ è sostituito con una sua stima. Tale funzione e la sua rispettiva trasformata logaritmica sono chiamate di **verosimiglianza** e **log-verosimiglianza profilo**.

$$L_p(\tau) = L(\tau, \hat{\zeta}_\tau) \qquad \text{e} \qquad l_p(\tau) = l(\tau, \hat{\zeta}_\tau) \tag{1.7}$$

La stima per $\hat{\zeta}_\tau$ è ottenuta come soluzione dell'equazione $\frac{\partial l(\tau,\zeta)}{\partial \zeta} = 0$, ovvero ponendo pari a 0 la derivata parziale in ζ della funzione di verosimiglianza propria considerando τ fissato. Per questo motivo è evidente che la stima di massima verosimiglianza profilo per τ coincide con quella di verosimiglianza propria.

Viene ora presentata una proprietà della verosimiglianza profilo di cui ci si servirà in seguito. Definita l'informazione osservata profilo come

$$j_p(\tau) = -\frac{\partial^2}{\partial\tau\partial\tau^T} l_p(\tau) = -\frac{\partial^2}{\partial\tau\partial\tau^T} l(\tau, \hat{\zeta}_\tau),$$

si può dimostrare che l'inversa di $j_p(\tau)$ è pari all'inversa della matrice di informazione osservata complessiva, nel blocco (τ, τ). Quindi se $j(\tau, \zeta)$ è una matrice a blocchi definita come

$$j(\tau, \zeta) = \begin{bmatrix} j_{\zeta\zeta}(\tau, \zeta) & j_{\zeta\tau}(\tau, \zeta) \\ j_{\tau\zeta}(\tau, \zeta) & j_{\tau\tau}(\tau, \zeta) \end{bmatrix},$$

allora vale che

$$j_p(\tau)^{-1} = [j(\tau, \hat{\zeta})^{-1}]_{\tau\tau} = \Big(j_{\tau\tau}(\tau, \hat{\zeta}) - j_{\tau\zeta}(\tau, \hat{\zeta}) j_{\zeta\zeta}(\tau, \hat{\zeta})^{-1} j_{\zeta\tau}(\tau, \hat{\zeta})\Big)^{-1}, \tag{1.8}$$

il cui risultato deriva dalla regola di inversione a blocchi.

Per sistemi d'ipotesi in cui si ha che $H_0 : \tau = \tau_0$ e $H_1 : \tau \neq \tau_0$, esiste una statistica test detta di **log-rapporto di verosimiglianza profilo**. Si può dimostrare che essa sotto H_0 converge in distribuzione ad una distribuzione χ^2_q, dove $q = \dim(\tau_0)$. La sua definizione è

$$W_p(\tau_0) = 2\big(l_p(\hat{\tau}) - l_p(\tau_0)\big) = 2\big(l(\hat{\tau}, \hat{\zeta}_\tau) - l(\tau_0, \hat{\zeta}_{\tau_0})\big).$$

1.7 Metodo di Newton-Raphson

Alcune difficoltà sorgono quando si tenta di ottenere la soluzione di $l_*(\theta) = 0$. Può accadere che non sia possibile esplicitare la stima di massima verosimiglianza ed in tal caso si deve ricorrere ad algoritmi numerici. Uno di questi,

che non necessariamente è il più efficiente in termini computazionali, è il metodo di Newton-Raphson. È un algoritmo iterativo che necessita di un punto d'inizio. Ne viene data, in questo paragrafo, una sommaria spiegazione.

Lo sviluppo in serie di Taylor della funzione $l_*(\theta)$ porge

$$l_*(\theta) = l_*(\theta_0) + l_{**}(\theta_0)(\theta - \theta_0). \tag{1.9}$$

Imponendo la soluzione $l_*(\theta) = 0$ e ricordando che $l_{**}(\theta) = -j(\theta)$ si ottiene che

$$\begin{aligned} &l_*(\theta_0) - j(\theta_0)(\theta - \theta_0) = 0 \\ &j(\theta_0)(\theta - \theta_0) = l_*(\theta_0) \\ &\theta = \theta_0 + j(\theta_0)^{-1} l_*(\theta_0), \end{aligned}$$

da cui è possibile stabilire un algoritmo iterativo in cui

$$\hat{\theta}_{k+1} = \hat{\theta}_k + j(\hat{\theta}_k)^{-1} l_*(\hat{\theta}_k). \tag{1.10}$$

1.7.1 Algoritmi numerici con R

Nel corso della relazione verrà utilizzato il metodo di Newton-Raphson come approssimazione numerica della stima di massima verosimiglianza. È bene tener conto che, tuttavia, esistono numerosi altri algoritmi che permetterebbero di ottenere lo stesso risultato. Molti di questi sono già implementati nel software R. Per maggiori approfondiment riguardanti il software utilizzato si veda R Core Team (2012).

A titolo esemplificativo, vengono calcolate le stime di massima verosimiglianza per i dati riportati in Tabella 1.1, che rappresentano il tempo di rottura di alcune molle, sottoposte a ripetuti sforzi tramite pesi differenti. I dati hanno la seguente struttura. A questo scopo, si utilizzeranno solamente le prime 10 osservazione relative alla variabile `cycles` che si assumerà siano realizzazioni indipendenti provenienti da una variabile Weibull che abbia funzione di densità pari a

$$p(x; \gamma, \lambda) = \left(\frac{\gamma}{\lambda}\right)\left(\frac{x}{\lambda}\right)^{\gamma-1} \exp\{-\left(\frac{x}{\lambda}\right)^{\gamma}\}.$$

Tabella 1.1: Le prime 10 osservazioni dei dati `stress`

	cycles	cens	stress
1	225	1	950
2	171	1	950
3	198	1	950
4	189	1	950
5	189	1	950
6	135	1	950
7	162	1	950
8	135	1	950
9	117	1	950
10	162	1	950

```
y<- stress$cycles[1:10]
```

Si definisce quindi la funzione di log-verosimiglianza cambiata di segno

```
nlog.weibull<-function(par,y) -sum(dweibull(y,par[1],par[2],log=TRUE))
```

Quindi si utilizza il comando `nlminb`, che permette la minimizzazione di una funzione

```
smv<- nlminb (start=c(0.5,0.5), nlog.weibull, lower=c(1e-8,1e-8),y=y)
```

La stima di massima verosimiglianza risulta essere $\hat{\theta} = (\hat{\gamma}, \hat{\lambda}) = (5.97, 181.4)$. Inoltre, tramite la libreria `nlme` è possibile ottenere la matrice hessiana e, dunque, anche quella di informazione osservata. Utilizzando lo stesso campione, il codice da utilizzare è

```
library(nlme)
theta.cappello<-smv$par
fdHess(theta.cappello, nlog.weibull, y=y)
```

Alternativamente si può utilizzare la libreria `numDeriv`, che permette di ottenere lo stesso risultato con algoritmi più affidabili. Il codice utilizzato è

```
library(numDeriv)
theta.cappello<-smv$par
hessian(nlog.weibull, theta.cappello, y=y)
```

Trattandosi di un caso piuttosto semplice, entrambe le funzioni portano a risultati estremamente simili. La matrice di informazione osservata è pari a

$$j(\hat{\theta}) = \begin{bmatrix} 0.524 & -0.024 \\ -0.024 & 0.0108 \end{bmatrix}.$$

Questo risultato è stato mostrato non tanto perché di interesse in sè, ma perché la procedura che si è utilizzata verrà ripresa successivamente. I valori che si ottengono numericamente soffrono di un certo grado di approssimazione ma sono molto affidabili. I risultati analitici, invece, sono matematicamente più precisi ma possono contenere errori di distrazione o logici, commessi dalla persona che li ha ottenuti.

Oltre a queste considerazioni, si aggiunge il fatto che spesso non è proprio possibile giungere ad un risultato analitico. Non sempre l'integrazione di una funzione porta ad una primitiva esprimibile come combinazione di funzioni elementari, si pensi ad esempio a $\Phi(y)$. In questi casi il calcolo numerico quindi non è più una valida alternativa, ma l'unica via percorribile.

Infine, il metodo numerico non richiede grandi sforzi per essere applicato. È una procedura automatica che necessita unicamente della funzione di log-verosimiglianza.

Capitolo 2

Modello lineare e trasformazione di variabili

2.1 Introduzione

Nel capitolo precedente si è discusso in maniera molto generale di *modelli statistici parametrici*. Una loro sottoclasse è data dai **modelli di regressione lineare** la cui importanza è indiscussa in ambito econometrico, sociale, medico e non soltanto. L' origine degli studi riguardanti i modelli lineari è abbastanza incerta: le prime fondamentali nozioni vennero introdotte da Adrien-Marie Legendre e Carl Friedrich Gauss, nei primi anni del 1800. Esse tuttavia si svilupparono notevolmente soprattutto quando si cominciò ad ipotizzare una distribuzione per gli errori, estensione ad opera di Karl Pearson e George Udny Yule tra il 1897 ed il 1903. Si veda *Wikipedia, Modello Lineare* (2013) e i documenti ivi citati.

Oggi i modelli lineari sono ampiamente utilizzati e studiati. Ne sono state introdotte numerose estensioni: una fra tutte sono i **modelli lineari generalizzati**, che però non verranno trattati in questa relazione. Per una loro esaustiva trattazione si veda McCullagh e Nelder (1989). Queste innovazioni sono state possibili grazie all'evoluzione della tecnologia che ha permesso lo sviluppo di rapidi algoritmi di calcolo numerico.

Nei modelli lineari vengono fatte alcune ipotesi circa la natura delle os-

servazioni e spesso esse vengono contraddette in fase di verifica. In questi casi si può scegliere di ignorare queste violazioni, se si ritiene che siano lievi o irrilevanti, oppure modificare il modello di partenza, nel tentativo di correggerle. Un metodo diffuso consiste nell'agire sulla variabili, dipendenti o indipendenti che siano, trasformandole con opportune funzioni. Nel caso si decidesse di trasformare la variabile dipendente, sono state fatte numerose proposte di cui la soluzione di Box-Cox rappresenta una generalizzazione. In questo capitolo verranno introdotti alcuni concetti fondamentali riguardanti i modelli lineari e verrà discusso l'approccio proposto da Box-Cox, riportando alcuni esempi.

2.2 Il modello lineare

Si definisce modello lineare una qualunque relazione esprimibile nella forma

$$Y = X\beta + \varepsilon, \tag{2.1}$$

in cui X è una matrice di dimensione $n \times p$ nella quale ciascuna colonna rappresenta una variabile, β è un vettore di dimensione $p \times 1$ di parametri ignoti ed ε è un vettore stocastico di dimensione $p \times 1$. Le caratteristiche del modello lineare quindi sono

- la componente stocastica viene addizionata al resto del modello;
- la funzione che esprime Y in funzione di X è lineare nei parametri.

Un problema di regressione lineare consiste nel cercare di stimare i parametri ignoti contenuti nel vettore β a partire dalle realizzazioni di $y = (y_1, \ldots, y_n)$ che si suppone abbiano la struttura descritta nell'equazione (2.1).

Senza ulteriori restrizioni la stima di β è praticamente impossibile ed è per questo che si suppone che valgano almeno le **ipotesi del second'ordine**. Si richiede cioè che

- X sia una matrice non stocastica di rango pieno: $\text{rank}(X) = p$;
- la media degli errori sia nulla: $\mathbf{E}[\varepsilon] = 0$;

- la varianza degli errori sia costante (**omoschedasticità**) e questi siano incorrelati tra loro: $\text{Var}(\varepsilon) = \sigma^2 I_n$.

Utilizzando il criterio dei minimi quadrati è possibile giungere a stimatori consistenti per β che godono di ottime proprietà. Tuttavia per condurre analisi inferenziali bisogna ipotizzare una distribuzione per le osservazioni. Sotto l' **ipotesi di normalità** si ha che ε è distribuito come una normale multivariata e quindi

$$\varepsilon \sim N_n(0, \sigma^2 I_n) \qquad \text{che implica} \qquad Y \sim N_n(X\beta, \sigma^2 I_n).$$

2.2.1 La stima dei parametri

Le stime per β, tramite il criterio dei minimi quadrati, si ottengono minimizzando la funzione $Q(\beta) = (Y - X\beta)^T(Y - X\beta) = ||Y - X\beta||^2$, cioè la distanza tra i valori assunti da Y e quelli previsti dal modello. Si può dimostrare che nella funzione di massima verosimiglianza profilo per β, sotto l'ipotesi di normalità, il punto di massimo assoluto coincide con quello di minimo della funzione $Q(\beta)$. Entrambi i metodi di stima per β portano dunque al risultato

$$\hat{\beta} = (X^T X)^{-1} X^T Y.$$

Sotto le condizioni del second'ordine lo stimatore si verifica essere non distorto e consistente ed inoltre è denominato **BLUE** (Best Linear Unbiased Estimator). Ciò significa che esso è lo stimatore più efficiente nella classe degli stimatori lineari non distorti. Questo risultato è stato dimostrato nel noto **teorema di Gauss-Markov**. Si veda Pace e Salvan (2001, pag. 296). Una stima per σ^2 è

$$\hat{\sigma}^2 = \frac{||y - X\hat{\beta}||^2}{n},$$

che è la stima di massima verosimiglianza. In genere però, poichè $\hat{\sigma}^2$ è uno stimatore distorto, si preferisce una sua lieve correzione: s^2, definita come

$$s^2 = \frac{||y - X\hat{\beta}||^2}{n - p},$$

che è uno stimatore corretto della varianza del modello.

2.2.2 Inferenza sui parametri

La semplice stima dei parametri sarebbe inutile se non si riuscisse in qualche modo a quantificarne la precisione. Perciò si ottengono dei test che sono connessi alla verosimiglianza e sono però propri solo dei modelli lineari normali.

Si voglia verificare un sistema d'ipotesi del tipo $H_0 : H\beta = 0$ e $H_1 : H\beta \neq 0$ dove H è una matrice $q \times p$, con $q \leq p$. Ciascuna riga rappresenta uno dei vincoli lineari che si intende imporre. Nel capitolo precedente è stato definito il test statistico log-rapporto di verosimiglianza che permetterebbe di verificare H_0. La distribuzione normale rende i calcoli analitici piuttosto agevoli e questo consente di arrivare, tramite opportune trasformazioni monotone di $W_p(\theta)$, ad una quantità la cui distribuzione è completamente nota. Si dimostra che

$$F = \frac{||\hat{\mu} - \hat{\mu}_0||^2/q}{||y - \hat{\mu}||^2/(n-p)} \sim F_{(q,n-p)},$$

in cui $\hat{\mu} = X\hat{\beta}$ e $\hat{\mu}_0 = X\hat{\beta}_0$. Nel caso in cui $q = 1$ si ha che $t^2 = F$, con t che ha distribuzione t di student con $n - p$ gradi di libertà. Questo implica che con un unico test è possibile verificare tutti i vincoli lineari che si desiderano. Esistono alcuni rilevanti casi particolari: spesso si vuole verificare la nullità di un parametro o di un intero gruppo di questi. Se l'ipotesi nulla venisse accettata, si potrebbero escludere alcune variabili dal modello con un risparmio dal punto di vista interpretativo e, a volte, anche economico.

Queste interessanti applicazioni si reggono sulle ipotesi sopra formulate e, se queste venissero a cadere, i test statistici potrebbero diventare privi di significato. Ecco perchè verranno presentate alcune soluzioni nel paragrafo successivo.

2.3 Trasformazioni della variabile dipendente

Le ottime proprietà del modello lineare normale decadono se qualcuna delle ipotesi iniziali non è rispettata. Ad esempio, i dati potrebbero non avere varianza costante (**eteroschedasticità**), oppure gli errori potrebbero

non provenire da una distribuzione normale o essere correlati. Non esiste un unico approccio risolutivo e le vie percorribili in generale sono:

- ignorare la violazione delle ipotesi e trattare il modello come se queste fossero vere;
- inserire nel modello nuove variabili. È bene far notare che spesso questo non è possibile per motivi economici o di contesto;
- rispecificare il modello. In presenza di errori correlati, ad esempio, potrebbe essere conveniente utilizzare modelli per serie storiche;
- trasformare le variabili tramite una funzione

Non esiste una ricetta che permetta di scegliere la soluzione migliore, ammesso che ne esista una sola. La scelta dipende dalle esigenze di chi poi utilizzerà il modello e dalla gravità della violazione delle ipotesi.

Anche nel caso in cui si decidesse di trasformare una variabile, le possibilità sono numerose. Si potrebbe agire contemporaneamente su tutte le variabili o su una sola di esse. Focalizzando l'attenzione sulla **variabile dipendente**, le funzioni utilizzate più di frequente sono: *la radice quadrata, il logaritmo naturale, il reciproco.*

Una prima **classe** di trasformazioni per la dipendente è stata proposta in Tukey (1957). Essa prevedeva

$$y_\lambda = \begin{cases} y^\lambda & \text{se } \lambda \neq 0 \\ \log y & \text{se } \lambda = 0 \end{cases}, \tag{2.2}$$

per valori di y positivi. A seconda del parametro λ si possono ottenere una grande varietà di trasformazioni. Due raffinamenti della (2.2) sono dati dalle **trasformazioni Box-Cox**, in Box e Cox (1964), le cui espressioni sono

$$y_\lambda = \begin{cases} \frac{y^\lambda - 1}{\lambda} & \text{se } \lambda \neq 0 \\ \log y & \text{se } \lambda = 0 \end{cases}, \tag{2.3}$$

per valori di y strettamente positivi e

$$y_\lambda = \begin{cases} \frac{(y+\lambda_2)^{\lambda_1} - 1}{\lambda_1} & \text{se } \lambda_1 \neq 0 \\ \log(y + \lambda_2) & \text{se } \lambda_1 = 0 \end{cases}, \tag{2.4}$$

per valori di y maggiori di λ_2. Nel seguito della relazione si farà riferimento alla (2.3) a meno che non venga espressamente detto il contrario. Si noti che essa è pari alla (2.2) a meno di trasformazioni lineari. Questo implica che i due modelli sono del tutto equivalenti. Tuttavia vale l'apprezzabile proprietà

$$\lim_{\lambda \to 0} \frac{y^\lambda - 1}{\lambda} = \log y,$$

che implica la continuità della funzione in λ.

Molte altre trasformazioni sono state proposte nel corso degli anni, ciascuna per correggere difetti della trasformazione Box-Cox. Per una rassegna esaustiva si veda Sakia (1992). Nonostante ciò, la sua semplicità ha fatto in modo che si diffondesse rapidamente e divenisse uno standard nell'ambito della trasformazione di variabili.

Una formulazione alternativa di una certa rilevanza è presente in Bickel e Doksum (1981, pag. 297), riportata qui di seguito

$$y_\lambda = \frac{\operatorname{sgn}(y)|y|^\lambda - 1}{\lambda}, \tag{2.5}$$

con $\lambda > 0$. Per gli y positivi la trasformazione, ed anche la funzione di verosimiglianza, coincide con la (2.3). Tuttavia ora y può assumere **anche valori negativi**. Questa trasformazione è, da un punto di vista teorico, più corretta perchè il campo di variazione di y_λ è ora l'intero insieme reale.

2.4 Modalità d'utilizzo

L'approccio usuale consiste nello stimare il parametro λ, spesso tramite verosimiglianza, per poi trattare la stima, o un suo intero vicino, **come noto**. Sono gli stessi Box e Cox a proporre questa modalità in Box e Cox (1964, pag. 239). Si deve però stabilire se è d'interesse studiare il fenomeno in una determinata scala, suggerita dai dati, oppure lo si vuole analizzare in una scala ignota che dipende dal vero parametro λ_0, come loro stessi fanno notare. Nel secondo caso si sta quindi supponendo che il modello correttamente specificato sia

$$\frac{Y^\lambda - 1}{\lambda} = X\beta + \varepsilon. \tag{2.6}$$

È evidente che non si può trascurare la variabilità aggiunta che la stima di λ comporta. In particolare le varianze associate alle stime dei parametri β saranno presumibilmente maggiori di quelle del modello in cui λ è trattato come noto. Si dovrà quindi utilizzare un approccio alternativo per cercare di quantificare la variabilità aggiuntiva.

2.5 I dati cars

Verrà ora presentato un dataset per illustrare la procedura originaria proposta da Box-Cox. Si è scelto un insieme di dati in cui, ovviamente, vi fosse la necessità di una trasformazione della variabile dipendente che fosse, inoltre, strettamente positiva.

I dati *cars* sono relativi alla distanza percorsa da un auto, che viaggiava ad una certa velocità prima di fermarsi. La distanza è espressa in piedi, la velocità in miglia orarie. Sono 50 osservazioni che risalgono agli anni venti. Sono presenti nel software R e possono essere facilmente richiamati col comando `data(cars)`. In Figura 2.1 v'è una loro rappresentazione. Si è interessati a spiegare la distanza percorsa dalle auto in funzione della loro velocità.

Appare evidente che intercorre una relazione tra le due variabili. Si intravede tuttavia un problema di eteroschedasticità nel caso si scegliesse di utilizzare semplicemente un modello lineare, anche se in forma non eccessivamente grave.

Si introduce allora la trasformazione Box-Cox e dunque il modello viene definito in questa maniera

$$\frac{Y_i^{\lambda} - 1}{\lambda} = \beta_0 + \beta_1 x_i + \varepsilon_i,$$

per $i = 1, \ldots, 50$, in cui sono valide tutte le ipotesi fatte in questo capitolo. La stima di massima verosimiglianza è $\hat{\lambda} = 0.43$, ottenuta con i metodi descritti nei capitoli successivi. Al posto di utilizzare la stima di massima verosimiglianza si assume che λ sia noto e posto pari a 0.5. Il modello diventa

$$\sqrt{Y_i} = \beta_0 + \beta_1 x_i + \varepsilon_i, \tag{2.7}$$

Utilizzando il software R il modello viene stimato e sono riportate in Tabella 2.1 le stime dei coefficienti con i relativi scarti quadratici medi.

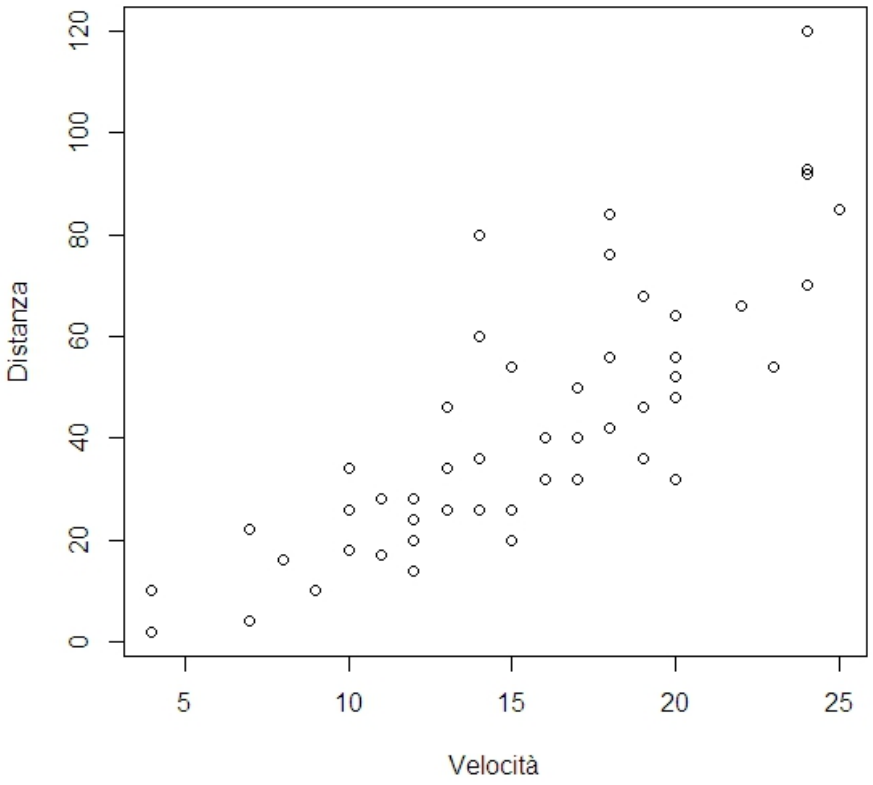


Figura 2.1: I dati *cars*

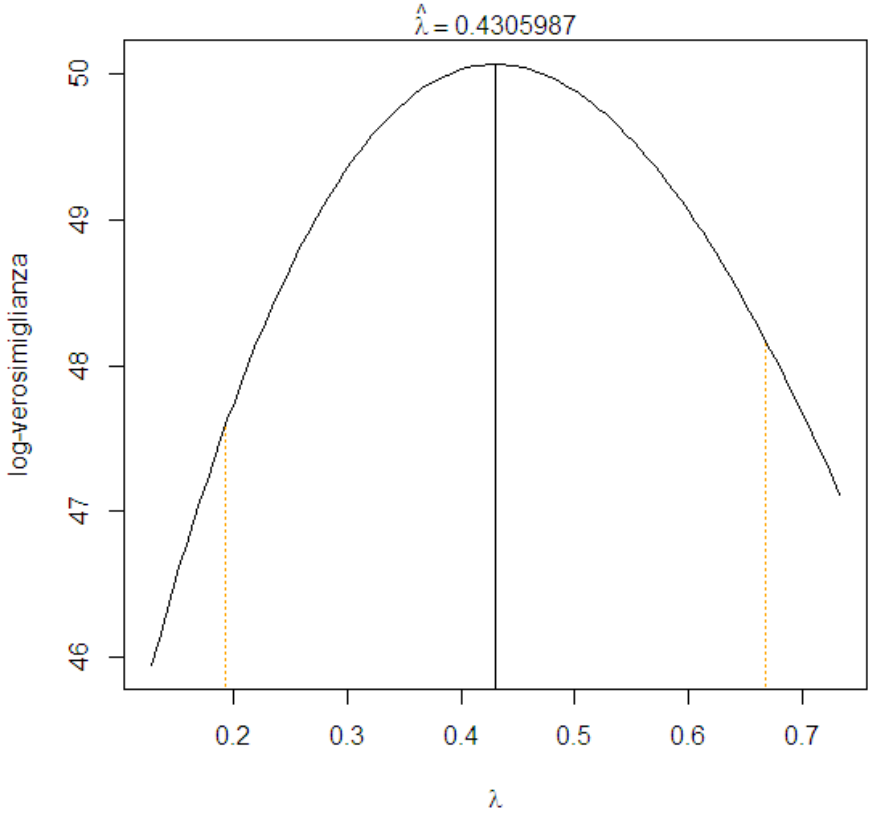


Figura 2.2: Log-verosimiglianza profilo per λ nei dati *cars*

Tabella 2.1: Sommario di R per la stima del modello $\sqrt{Y_i} = \beta_0 + \beta_1 x_i + \varepsilon_i$

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	1.2771	0.4844	2.64	0.0113
speed	0.3224	0.0298	10.83	0.0000

L'adattamento è buono, non si riscontrano particolari problemi nei grafici diagnostici. Questi stessi dati verranno rianalizzati successivamente, utilizzando un approccio differente. Ciò che per ora si fa notare è che l'unica variabile esplicativa è indubbiamente significativa.

Capitolo 3

Box-Cox e l'approccio di verosimiglianza

3.1 Introduzione

Per condurre un'analisi approfondita circa gli effetti di λ sul resto del modello è necessario ricavare le matrici di informazione osservata $j(\theta)$ e quella di informazione attesa $I(\theta)$ da utilizzare, ad esempio, nel test statistico definito nella (1.5). Grazie a queste due matrici sarà possibile quindi avere un'evidenza empirica a sostegno della supposizione dell'aumento della variabilità nel modello. La prima può essere ottenuta sia analiticamente che numericamente. La seconda invece comporta alcune difficoltà aggiuntive. Per ottenerle si dovrà innanzitutto costruire un algoritmo di approssimazione numerica con il quale ottenere le stime di massima verosimiglianza. Si è deciso di mostrare ogni singolo passaggio effettuato per raggiungere il risultato. I risultati inerenti alla matrice di informazione osservata e l'algoritmo di Newton-Raphson sono parzialmente basati su Scott, 1999.

Si ipotizza che il modello abbia la forma specificata nella (2.6), che gli errori siano normali e che valgano le ipotesi del second'ordine.

3.2 Funzione di verosimiglianza

Il primo passo consiste nel ricavare la funzione di densità per Y, da cui poi si otterrà la funzione di verosimiglianza e tutte le quantità ad essa collegate. La distribuzione di Y non è nota ma può essere calcolata analiticamente. Sia Y_λ la variabile trasformata per la quale vale che

$$Y_\lambda = X\beta + \varepsilon.$$

Allora seguirà che $Y_\lambda \sim N(X\beta, \sigma^2 I_n)$ e la funzione di densità sarà pari a

$$f_{y_\lambda}(y_\lambda) = \frac{1}{(2\pi\sigma)^{n/2}} e^{-\frac{1}{2\sigma^2}(y_\lambda - X\beta)^T(y_\lambda - X\beta)},$$

che si ricava dalla definizione di normale multivariata. Non è questa tuttavia la quantità d'interesse: si vuole trovare la funzione di densità per Y. Si definisca $Y_\lambda = g(Y)$ la relazione che intercorre tra le due variabili. È noto che le funzioni di densità sono legate dalla relazione

$$f_y(y) = f_{y_\lambda}\big(g(y)\big)|J(y)|, \tag{3.1}$$

dove $|J(y)|$ è il determinante della matrice jacobiana definita come $J(y) = \frac{\partial g(y)}{\partial y}$. Si ottiene che

$$\frac{g(y_i)}{\partial y_j} = 0 \quad \text{per } i \neq j \qquad \text{e} \qquad \frac{g(y_i)}{\partial y_j} = y_j^{\lambda-1} \quad \text{per } i = j,$$

con $i, j = 1, \ldots, n$. Quindi risulta che $J(y) = \text{diag}\big(y_1^{\lambda-1}, \ldots, y_n^{\lambda-1}\big)$. Trattandosi di una matrice diagonale, il determinante è semplicemente il prodotto di ciascun elemento:

$$|J(y)| = \prod_{i=1}^{n} y_i^{\lambda-1}.$$

A questo punto, sfruttando la (3.1), si ottiene la funzione di densità e dunque anche la funzione di verosimiglianza per Y. Viene riportata solamente la seconda che è pari a

$$L(\theta; y) = \frac{1}{(2\pi\sigma)^{n/2}} e^{-\frac{1}{2\sigma^2}(y_\lambda - X\beta)^T(y_\lambda - X\beta)} \prod_{i=1}^{n} y_i^{\lambda-1},$$

dove $\theta = (\beta, \sigma^2, \lambda)$. Come di consueto, si preferirà non trattare direttamente la funzione di verosimiglianza, ma la sua trasformazione logaritmica. Perciò si proseguirà utilizzando la funzione di log-verosimiglianza che è

$$l(\theta; y) = -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}(y_\lambda - X\beta)^T(y_\lambda - X\beta) + (\lambda - 1)\sum_{i=1}^{n}\log y_i.$$

3.3 Funzione punteggio

La funzione punteggio, definita nella (1.1), verrà calcolata a gruppi di parametri. Si procede innanzitutto alla derivazione per β e per σ^2, il cui calcolo è abbastanza agevole. Derivando per β si ottiene che

$$\begin{aligned}
\frac{\partial l(\theta; y)}{\partial \beta} &= \partial \frac{-\frac{1}{2\sigma^2}(y_\lambda - X\beta)^T(y_\lambda - X\beta)}{\partial \beta} \\
&= -\frac{1}{2\sigma^2}\partial \frac{(y_\lambda^T y_\lambda - (X\beta)^T y_\lambda - y_\lambda^T X\beta + \beta^T X^T X\beta)}{\partial \beta} \\
&= -\frac{1}{2\sigma^2}(-2X^T y_\lambda + 2X^T X\beta) \\
&= -\frac{1}{\sigma^2}(-X^T y_\lambda + X^T X\beta).
\end{aligned}$$

Inoltre la derivazione per σ^2 fornisce

$$\frac{\partial l(\theta; y)}{\partial \sigma^2} = -\frac{n}{2\sigma^2} + \frac{(y_\lambda - X\beta)^T(y_\lambda - X\beta)}{2(\sigma^2)^2}.$$

Si nota subito che queste due funzioni sono molto simili alle corrispondenti quantità di verosimiglianza in cui, al posto della variabile y, c'è la sua trasformazione y_λ. Ciò comporta che si otterranno le consuete stime di massima verosimiglianza per la variabile trasformata, per $\hat{\lambda}$ fissato.

Infine si deriva rispetto a λ, ottenendo

$$\begin{aligned}
\frac{\partial l(\theta; y)}{\partial \lambda} &= -\frac{1}{2\sigma^2}\partial \frac{(y_\lambda - X\beta)^T(y_\lambda - X\beta)}{\partial \lambda} + \sum_{i=1}^{n}\log y_i \\
&= -\frac{2(y_\lambda - X\beta)^T}{2\sigma^2}\frac{\partial y_\lambda}{\partial \lambda} + \sum_{i=1}^{n}\log y_i \\
&= -\frac{u^T(y_\lambda - X\beta)}{\sigma^2} + \sum_{i=1}^{n}\log y_i,
\end{aligned}$$

dove $u = \frac{\partial y_\lambda}{\partial \lambda}$. Il generico elemento u_i per $\lambda \neq 0$ è dato da

$$u_i = \frac{\partial}{\partial \lambda} \frac{y_i^\lambda - 1}{\lambda} = \frac{y_i^\lambda \big(\log y_i^\lambda - 1\big) + 1}{\lambda^2}.$$

Per $\lambda = 0$ invece la funzione è ricavata come soluzione del limite $\lim_{\lambda \to 0} u_i$. Nel complesso risulta quindi che

$$u_i = \begin{cases} \frac{y_i^\lambda (\log y_i^\lambda - 1) + 1}{\lambda^2} & \text{se } \lambda \neq 0 \\ \frac{\log^2 (y_i)}{2} & \text{se } \lambda = 0 \end{cases}.$$

Riassumendo, la funzione score è il vettore seguente

$$l_*(\theta; y) = \begin{bmatrix} \frac{\partial l(\theta;y)}{\partial \beta} \\ \\ \frac{\partial l(\theta;y)}{\partial \sigma^2} \\ \\ \frac{\partial l(\theta;y)}{\partial \lambda} \end{bmatrix} = \begin{bmatrix} -\frac{1}{\sigma^2}(-X^T y_\lambda + X^T X \beta) \\ \\ -\frac{n}{2\sigma^2} + \frac{(y_\lambda - X\beta)^T (y_\lambda - X\beta)}{2(\sigma^2)^2} \\ \\ -\frac{u^T (y_\lambda - X\beta)}{\sigma^2} + \sum_{i=1}^n \log y_i \end{bmatrix}.$$

3.4 Matrice di informazione osservata

La matrice $j(\theta)$ è ricavata prima ancora di impostare le equazioni di verosimiglianza perchè sarà necessaria in seguito per il loro calcolo. Questa inversione nella procedura è quindi, in questo caso, concettualmente più lineare di quella usuale.

Come per la funzione $l_*(\theta; y)$, anche $j(\theta)$ verrà calcolata per gruppi di parametri. Si suddivide la matrice simmetrica $j(\theta)$ a blocchi, come segue

$$j(\theta) = \begin{bmatrix} j_{11}(\theta) & j_{12}(\theta) & j_{13}(\theta) \\ - & j_{22}(\theta) & j_{23}(\theta) \\ - & - & j_{33}(\theta) \end{bmatrix} = - \begin{bmatrix} \frac{\partial l(\theta;y)}{\partial \beta^T \partial \beta} & \frac{\partial l(\theta;y)}{\partial \sigma^2 \partial \beta} & \frac{\partial l(\theta;y)}{\partial \lambda \partial \beta} \\ - & \frac{\partial l(\theta;y)}{\partial (\sigma^2)^2} & \frac{\partial l(\theta;y)}{\partial \sigma^2 \partial \lambda} \\ - & - & \frac{\partial l(\theta;y)}{\partial \lambda^2} \end{bmatrix}. \tag{3.2}$$

Innanzitutto si ottiene il blocco per la derivata seconda di β

$$\begin{aligned} j_{11}(\theta) = -\frac{\partial l(\theta; y)}{\partial \beta^T \partial \beta} &= -\frac{\partial}{\partial \beta}\Big(-\frac{1}{\sigma^2}(-X^T y_\lambda + X^T X \beta)\Big) \\ &= \frac{X^T X}{\sigma^2}, \end{aligned}$$

ovvero una matrice di dimensione $p \times p$. Il vettore misto in β e σ^2 è

$$\begin{aligned} j_{12}(\theta) = -\frac{\partial l(\theta; y)}{\partial \sigma^2 \partial \beta} &= -\frac{\partial}{\partial \sigma^2}\Big(-\frac{1}{\sigma^2}(-X^T y_\lambda + X^T X\beta)\Big) \\ &= \frac{X^T y_\lambda - X^T X\beta}{(\sigma^2)^2}, \end{aligned}$$

di dimensione $p \times 1$. Il vettore misto in β e λ è

$$\begin{aligned} j_{13}(\theta) = -\frac{\partial l(\theta; y)}{\partial \lambda \partial \beta} &= -\frac{\partial}{\partial \lambda}\Big(-\frac{1}{\sigma^2}(-X^T y_\lambda + X^T X\beta)\Big) \\ &= -\frac{X^T u}{\sigma^2}, \end{aligned}$$

ricordando che si è definito $u = \frac{\partial y_\lambda}{\partial \lambda}$. Quindi $j_{13}(\theta)$ ha dimensione $p \times 1$. Lo scalare relativo alla derivata seconda per σ^2 è dato da

$$\begin{aligned} j_{22}(\theta) = -\frac{\partial l(\theta; y)}{\partial \sigma^2 \partial \sigma^2} &= -\frac{\partial}{\partial \sigma^2}\Big(-\frac{n}{2\sigma^2} + \frac{(y_\lambda - X\beta)^T(y_\lambda - X\beta)}{2(\sigma^2)^2}\Big) \\ &= -\frac{n}{2(\sigma^2)^2} + \frac{(y_\lambda - X\beta)^T(y_\lambda - X\beta)}{(\sigma^2)^3}. \end{aligned}$$

Lo scalare relativo alla derivata mista in λ e σ^2 è pari a

$$\begin{aligned} j_{23}(\theta) = -\frac{\partial l(\theta; y)}{\partial \lambda \partial \sigma^2} &= -\frac{\partial}{\partial \lambda}\Big(\frac{(y_\lambda - X\beta)^T(y_\lambda - X\beta)}{2(\sigma^2)^2}\Big) \\ &= -\frac{2(y_\lambda - X\beta)^T}{2(\sigma^2)^2}\frac{\partial y_\lambda}{\partial \lambda} \\ &= -\frac{u^T(y_\lambda - X\beta)}{(\sigma^2)^2} = -\frac{(y_\lambda - X\beta)^T u}{(\sigma^2)^2} \end{aligned}$$

Infine va calcolata la derivata seconda in λ

$$\begin{aligned} j_{33}(\theta) = -\frac{\partial l(\theta; y)}{\partial \lambda^2} &= \frac{\partial}{\partial \lambda}\frac{u^T(y_\lambda - X\beta)}{\sigma^2} \\ &= \frac{u^T u + (y_\lambda - X\beta)^T v}{\sigma^2}, \end{aligned}$$

in cui $v = \frac{\partial u}{\partial \lambda}$. La sua derivata all'elemento i-esimo per $\lambda \neq 0$ è data da:

$$v_i = \frac{\partial u_i}{\partial \lambda} = \frac{y_i^\lambda[(\log(y_i^\lambda)^2 - 2\log(y_i^\lambda) + 2] - 2}{\lambda^3}.$$

Come con u_i il caso per $\lambda = 0$ è definito come il $\lim_{\lambda\to 0} v_i = \frac{\log(y_i)^3}{3}$. Questo risultato è mostrato in appendice. Riassumendo, quindi, la matrice di informazione osservata è pari a

$$j(\theta) = \begin{bmatrix} \frac{X^TX}{\sigma^2} & \frac{X^Ty_\lambda - X^TX\beta}{(\sigma^2)^2} & -\frac{X^Tu}{\sigma^2} \\ - & -\frac{n}{2(\sigma^2)^2} + \frac{(y_\lambda - X\beta)^T(y_\lambda - X\beta)}{(\sigma^2)^3} & -\frac{(y_\lambda - X\beta)^Tu}{(\sigma^2)^2} \\ - & - & \frac{u^Tu + (y_\lambda - X\beta)^Tv}{\sigma^2} \end{bmatrix}.$$

3.5 Stime di massima verosimiglianza

Nella ricerca della stima di massima verosimiglianza, che va cercata tra le soluzioni dell'equazione $l_*(\theta) = 0$, si incorre in un problema: non tutte le soluzioni sono ricavabili esplicitamente. In particolare, l'equazione relativa alla derivata in λ non ammette soluzione esplicita. Per le altre si ricava che

$$\hat{\beta}_\lambda = (X^TX)^{-1}X^TY_\lambda, \tag{3.3}$$

$$\hat{\sigma}^2_\lambda = \frac{Y_\lambda^T(I_n - P)Y_\lambda}{n} = \frac{(Y_\lambda - X\hat{\beta})^T(Y_\lambda - X\hat{\beta})}{n}. \tag{3.4}$$

La matrice P è di proiezione ed è definita come $P = X(X^TX)^{-1}X^T$. Questo suggerisce che per trovare il vettore $\hat{\theta}$ è sufficiente massimizzare la funzione di verosimiglianza profilo per λ e sostituire il punto di massimo nella (3.3) e nella (3.4). La log-verosimiglianza profilo è pari a

$$l_p(\lambda; y) = -\frac{n}{2}\log(2\pi\hat{\sigma}^2_\lambda) - \frac{1}{2\hat{\sigma}^2_\lambda}(y_\lambda - X\hat{\beta}_\lambda)^T(y_\lambda - X\hat{\beta}_\lambda) + (\lambda - 1)\sum_{i=1}^n \log y_i$$

che è equivalente a

$$l_p(\lambda; y) = -\frac{n}{2}\log(\hat{\sigma}^2_\lambda) + \lambda\sum_{i=1}^n \log y_i \tag{3.5}$$

in cui la stima vincolata $\hat{\sigma}^2_\lambda$ è data dalla (3.4). Per massimizzare la (3.5) si possono utilizzare numerosi metodi numerici. Verrà utilizzato l'algoritmo di

Newton-Raphson che verrà poi confrontato con altri algoritmi numerici, per verificarne la correttezza.

Innanzitutto viene calcolata la derivata di $l_p(\lambda; y)$ che coincide con $\frac{\partial l(\theta;y)}{\partial \lambda}$ in cui però i parametri β e σ^2 sono sostituiti con la loro stima di massima verosimiglianza. Si ottiene quindi che:

$$g(\lambda) = \frac{\partial l_p(\lambda; y)}{\partial \lambda} = -\frac{u^T(y_\lambda - X\hat{\beta})}{\hat{\sigma}^2} + \sum_{i=1}^{n} \log y_i \tag{3.6}$$

La soluzione di $g(\lambda) = 0$ non è disponibile in forma chiusa, perciò l'algoritmo iterativo sarà:

$$\lambda_{k+1} = \lambda_k + j_p(\lambda_k)^{-1} g(\lambda_k) \tag{3.7}$$

seguendo le indicazioni date nella (1.10).

3.5.1 La varianza per λ

La funzione $j_p(\lambda)$ andrebbe calcolata come $j_p(\lambda) = \frac{\partial l_p(\lambda;y)}{\partial \lambda^2}$, ma poichè $j(\theta)$ è nota, essa si può ricavare abbastanza velocemente usando la (1.8).

Innanzitutto si fa notare che $j_{12}(\lambda, \hat{\zeta}_{\hat{\lambda}}) = j_{21}(\lambda, \hat{\zeta}_{\hat{\lambda}})^T = 0$, con $\zeta = (\beta, \sigma^2)$. Infatti:

$$j_{12}(\lambda, \hat{\zeta}_{\hat{\lambda}}) = \frac{X^T y_\lambda - X^T X\hat{\beta}}{(\hat{\sigma}^2)^2} = \frac{X^T y_\lambda - X^T X(X^T X)^{-1} X^T y_\lambda}{(\hat{\sigma}^2)^2} = 0$$

. Anche $j_{22}(\lambda, \hat{\zeta}_{\hat{\lambda}})$ può essere semplificata e risulta che

$$\begin{aligned} j_{22}(\lambda, \hat{\zeta}_{\hat{\lambda}}) &= -\frac{n}{2(\hat{\sigma}^2)^2} + \frac{(y_\lambda - X\hat{\beta})^T (y_\lambda - X\hat{\beta})}{(\hat{\sigma}^2)^3} \\ &= -\frac{n}{2(\hat{\sigma}^2)^2} + \frac{n\hat{\sigma}^2}{(\hat{\sigma}^2)^3} = -\frac{n}{2(\hat{\sigma}^2)^2} + \frac{n}{(\hat{\sigma}^2)^2} \\ &= \frac{n}{2(\hat{\sigma}^2)^2} \end{aligned}$$

Perciò una nuova suddivisione in blocchi dà

$$j(\lambda, \hat{\zeta}_{\hat{\lambda}}) = \begin{bmatrix} j_{\zeta\zeta}(\lambda) & j_{\zeta\lambda}(\lambda) \\ j_{\lambda\zeta}(\lambda) & j_{\lambda\lambda}(\lambda) \end{bmatrix} = \left[\begin{array}{cc|c} \frac{X^T X}{\hat{\sigma}^2} & 0 & -\frac{X^T u}{\hat{\sigma}^2} \\ 0^T & \frac{n}{2(\hat{\sigma}^2)^2} & -\frac{(y_\lambda - X\hat{\beta})^T u}{(\hat{\sigma}^2)^2} \\ \hline -\frac{u^T X}{\hat{\sigma}^2} & -\frac{(y_\lambda - X\hat{\beta})^T u}{(\hat{\sigma}^2)^2} & \frac{u^T u + (y_\lambda - X\hat{\beta})^T v}{\hat{\sigma}^2}, \end{array}\right]$$

e quindi complessivamente si ha che

$$j_p(\lambda)^{-1} = \big[j(\lambda, \hat{\zeta}_{\hat{\lambda}})^{-1}\big]_{\lambda\lambda} = \big(j_{\lambda\lambda}(\lambda) - j_{\lambda\zeta}(\lambda) j_{\zeta\zeta}(\lambda)^{-1} j_{\zeta\lambda}(\lambda)\big)^{-1}.$$

Si vuole verificare il risultato di:

$$\begin{aligned}
\hat{\sigma}^2 j_{\lambda\zeta}(\lambda) j_{\zeta\zeta}(\lambda)^{-1} j_{\zeta\lambda}(\lambda) &= \begin{bmatrix} -u^T X & -\frac{y_\lambda^T (I_n - P)u}{\hat{\sigma}^2} \end{bmatrix} \begin{bmatrix} (X^T X)^{-1} & 0 \\ 0^T & \frac{2\hat{\sigma}^2}{n} \end{bmatrix} \begin{bmatrix} -X^T u \\ -\frac{y_\lambda^T (I_n - P)u}{\hat{\sigma}^2} \end{bmatrix} \\
&= \begin{bmatrix} -u^T X (X^T X)^{-1} & -\frac{2 y_\lambda^T (I_n - P)u}{n} \end{bmatrix} \begin{bmatrix} -X^T u \\ -\frac{y_\lambda^T (I_n - P)u}{\hat{\sigma}^2} \end{bmatrix} \\
&= u^T P u + 2 \frac{(y_\lambda^T (I_n - P)u)^2}{n \hat{\sigma}^2}.
\end{aligned}$$

Quindi nel complesso si ottiene che:

$$\begin{aligned}
j_p(\lambda) &= \frac{u^T u + (y_\lambda - X\hat{\beta})^T v - u^T P u}{\hat{\sigma}^2} - \frac{2}{n} \Big(\frac{y_\lambda^T (I_n - P)u}{\hat{\sigma}^2} \Big)^2 \\
&= \frac{u^T (I_n - P)u + y_\lambda^T (I_n - P)v}{\hat{\sigma}^2} - \frac{2}{n} \Big(\frac{y_\lambda^T (I_n - P)u}{\hat{\sigma}^2} \Big)^2. \qquad (3.8)
\end{aligned}$$

Ottenuto questo risultato bisogna affidarsi ad un calcolatore per ottenere la stima per λ. Un esempio di codice per il software R è dato in appendice.

3.6 Trasformazione Bickel e Docksum

Finora si è scelto di ignorare una particolarità della trasformazione Box-Cox: non si è tenuto conto del fatto che gli errori variano nell'intervallo reale mentre la trasformazione è definita solo nei reali positivi. Nella maggior parte dei casi questo ha poco senso, a meno che non si voglia sostenere che la variabile Y possa assumere anche valori complessi, cosa che nei casi pratici avviene di rado.

Condizionandosi al caso in cui ciascuna realizzazione di Y è positiva, si ottengono risultati ragionevoli sia nelle stime di massima verosimiglianza, sia nella matrice di informazione osservata, pur trattandosi di un approccio formalmente poco corretto.

Qualora si intendesse calcolare il valore atteso $\mathbf{E}(Y)$, però, ci si scontrerebbe con questa incogruenza formale. Il valore che si otterrebbe sarebbe

complesso, $\mathbf{E}(Y) \in \mathbb{C}$, e non sarebbe più nè utilizzabile nè interpretabile. Per risolvere questa difficoltà si sceglie di **abbandonare la trasformazione Box-Cox** e di utilizzare al suo posto quella proposta da Bickel e Doksum, 1981, definita nella (2.5).

Quanto descritto nei capitoli precedenti, tuttavia, non perde di validità nel caso in cui $y > 0$. La trasformazione Bickel-Docksum non deve essere vista come un approccio completamente differente, quanto come un' estensione della trasformazione Box-Cox dato che, **per y > 0**, queste **coincidono**. Perciò sia le stime di massima verosimiglianza che la stima della matrice di informazione osservata saranno le stesse. Nel caso generale, invece, la funzione di log-verosimiglianza è

$$l_{BD}(\theta; y) = -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}(y_\lambda - X\beta)^T(y_\lambda - X\beta) + (\lambda - 1)\sum_{i=1}^{n}\log|y_i|,$$

che quindi differisce da $l(\theta; y)$ solamente per un modulo e per il fatto che y_λ ora rappresenta la nuova trasformazione. Il determinante della matrice jacobiana è

$$|J(y)| = \prod_{i=1}^{n}|y_i|^{\lambda-1}.$$

I primi due blocchi della funzione score sono praticamente identici a quelli precedentemente ottenuti. Vi sono alcune distinzioni però nel termine $\frac{\partial l(\theta;y)}{\partial\lambda}$. Si ricava infatti che

$$\frac{\partial l_{BD}(\theta; y)}{\partial\lambda} = -\frac{u_*^T(y_\lambda - X\beta)}{\sigma^2} + \sum_{i=1}^{n}\log|y_i|,$$

in cui, con conti analoghi ai precedenti, si definisce l'elemento i-esimo del vettore u_*

$$u_{*i} = \frac{\partial y_{i\lambda}}{\partial\lambda} = \frac{\text{sign}(y_i)|y_i|^\lambda(\log(|y_i|^\lambda) - 1) + 1}{\lambda^2}.$$

Non è necessario ricavare la matrice di informazione osservata *ex novo*, ma è sufficiente un accorgimento: $j_{BD}(\theta)$ si ottiene sostituendo u_* al vettore u e v_* al vettore v all'interno della matrice $j(\theta)$. Si definisce $v_* = \frac{\partial u_*}{\partial\lambda}$, il cui elemento i-esimo è

$$v_i* = \frac{\partial u_i*}{\partial\lambda} = \frac{\text{sign}(y_i)|y_i|^\lambda[(\log(|y_i|^\lambda)^2 - 2\log(|y_i|^\lambda) + 2] - 2}{\lambda^3}.$$

3.7 Matrice di informazione attesa

Anche se $j_{BD}(\theta)$ ha proprietà apprezzabili, esistono validi motivi che spingono alla ricerca della matrice $I(\theta)$. Innanzitutto $I(\theta)$ è **definita positiva** in qualunque punto essa venga calcolata, mentre questo non è vero per $j_{BD}(\theta)$ che lo è, in generale, solo nel punto di massima verosimiglianza. Inoltre $I(\theta)^{-1}$ raggiunge il **limite inferiore di Cramer-Rao** che implica, sostanzialmente, una maggiore efficienza dello stimatore. La dimostrazione e l'illustrazione di questo risultato sono date in Azzalini, 2008, pag. 79.

Come si è visto, però, la trasformazione Box-Cox non si presta al calcolo dei valori attesi. Qui di seguito quindi verrà mostrato come ricavare la matrice di informazione attesa della trasformazione (2.5) basandosi parzialmente sui risultati già ottenuti in Bickel e Doksum, 1981.

3.7.1 Calcolo degli elementi di $I(\theta)$

La matrice $I(\theta)$ non può essere calcolata tramite un processo numerico automatico come avviene con $j_{BD}(\theta)$. Si suppone anche questa volta che $I(\theta)$ sia una matrice decomposta a blocchi

$$I(\theta) = \begin{bmatrix} i_{11}(\theta) & i_{12}(\theta) & i_{13}(\theta) \\ - & i_{22}(\theta) & i_{23}(\theta) \\ - & - & i_{33}(\theta) \end{bmatrix}, \tag{3.9}$$

e verrà calcolata come valore atteso di ciascun elemento di $j_{BD}(\theta)$. Per prima cosa si otterranno i valori attesi degli elementi che non presentano grosse problematiche computazionali. L'elemento per i coefficienti β è pari a

$$i_{11}(\theta) = \mathbf{E}\Big[\frac{X^T X}{\sigma^2}\Big] = \frac{X^T X}{\sigma^2}, \tag{3.10}$$

che quindi coincide con quello della matrice di informazione osservata, $i_{11}(\theta) = j_{11}(\theta)$. L'elemento in β e σ^2 è pari a

$$i_{12}(\theta) = \mathbf{E}\Big[\frac{X^T Y_\lambda - X^T X\beta}{(\sigma^2)^2}\Big] = \frac{X^T X\beta - X^T X\beta}{(\sigma^2)^2} = 0, \tag{3.11}$$

essendo $\mathbf{E}[Y_\lambda] = X\beta$. In questo caso $i_{12}(\theta) = j_{12}(\hat{\theta})$ ed inoltre anche $i_{21}(\theta) = j_{21}(\hat{\theta})$. L'elemento per σ^2 è pari a

$$i_{22}(\theta) = \mathbf{E}\Big[-\frac{n}{2(\sigma^2)^2} + \frac{(Y_\lambda - X\beta)^T(Y_\lambda - X\beta)}{(\sigma^2)^3}\Big].$$

Per definizione, posto $Q = Z^T Z$ con $Z \sim N_n(0, I_n)$, si ha che $Q \sim \chi^2_n$. Da ciò segue che se $U \sim N_n(0, \sigma^2 I_n)$ allora $U^T U \sim \sigma^2 \chi^2_n$. È noto che $\mathbf{E}[Q] = n$. Poichè è immediato notare che $(Y_\lambda - X\beta) \sim N_n(0, \sigma^2 I_n)$, si ottiene che $\mathbf{E}[(Y_\lambda - X\beta)^T(Y_\lambda - X\beta)] = n\sigma^2$. Perciò

$$i_{22}(\theta) = -\frac{n}{2(\sigma^2)^2} + \frac{n\sigma^2}{(\sigma^2)^3} = \frac{n}{2(\sigma^2)^2}. \tag{3.12}$$

Quindi anche per questo caso si ha che $i_{22}(\theta) = j_{22}(\hat{\theta})$. Per gli elementi successivi la questione diventa più complessa perchè non si riesce a ricondursi a distribuzioni note di cui si conosce il valore atteso. Si ricorrerà ad algoritmi numerici per approssimare i valori attesi che non sono ricavabili analiticamente.

Si introduce un risultato preliminare, dimostrato in Ross, 2007, pag. 204. Sia Z una variabile casuale continua con funzione di densità $f(z)$ e $g(\cdot)$ una funzione a valori reali; allora vale che

$$\mathbf{E}[g(Z)] = \int_{-\infty}^{+\infty} g(z) f(z)\, dz. \tag{3.13}$$

I valori attesi di cui non si dispone di un risultato analitico sono, sostanzialmente, tre. In particolare si vogliono conoscere

$$\begin{aligned} i_{31}(\theta) &= \mathbf{E}\Big[-\frac{u_*^T X}{\sigma^2}\Big], \\ i_{32}(\theta) &= \mathbf{E}\Big[-\frac{(Y_\lambda - X\beta)^T u_*}{(\sigma^2)^2}\Big], \\ i_{33}(\theta) &= \mathbf{E}\Big[\frac{u_*^T u_* + (Y_\lambda - X\beta)^T v_*}{\sigma^2}\Big]. \end{aligned} \tag{3.14}$$

Si procede quindi per integrazione numerica tramite la funzione `integrate` del software R, sfruttatando la (3.13). Si è scelto di utilizzare V, definita qui di seguito, come variabile di riferimento, la cui distribuzione è nota. È

necessario quindi esprimere ciascun termine in funzione di V. Si definisce quindi

$$V = \text{sign}(Y)|Y|^{\lambda} \qquad \text{che implica che} \qquad |V| = |Y|^{\lambda}. \tag{3.15}$$

Ne consegue inoltre che $V \sim N_n(1+\lambda X\beta, \lambda^2\sigma^2 I_n)$, grazie alle proprietà della distribuzione normale. I vettori u_* e v_* possono essere riscritti come segue

$$\begin{aligned} u_* &= \frac{1 + V(\log|V| - 1)}{\lambda^2}, \\ v_* &= \frac{V\big(\log(|V|)^2 - 2\log|V| + 2\big) - 2}{\lambda^3}. \end{aligned}$$

Ponendo μ pari a $X\beta$ si ottiene anche che

$$Y_\lambda - X\beta = \frac{V-1}{\lambda} - \mu. \tag{3.16}$$

Unendo queste quantità nella maniera indicata nella (3.14), si perviene al risultato finale. Non vengono riportate le formule complessive per evitare inutili appesantimenti nella lettura. Per $i_{32}(\theta)$ può essere utile ricordare che

$$\mathbf{E}\Big[-\frac{u_*^T(y_\lambda - X\beta)}{\sigma^2} + \sum_{i=1}^{n}\log|y_i|\Big] = 0,$$

per le proprietà del vettore score. Si ricava quindi che

$$i_{32}(\theta) = \mathbf{E}\Big[-\frac{(Y_\lambda - X\beta)^T u_*}{(\sigma^2)^2}\Big] = \frac{1}{\sigma^2\lambda}\sum_{i=1}^{n}\mathbf{E}[\log|V_i|]. \tag{3.17}$$

3.8 Alcune considerazioni

In questo capitolo sono state analizzate due differenti trasformazioni: quella proposta da Box-Cox e quella suggerita da Bickel e Docksum. Esse non sono intercambiabili e non possono essere ipotizzate contemporaneamente. Un approccio sensato, per dati positivi, potrebbe essere: ipotizzare la trasformazione (2.5) per la variabile dipendente tenendo conto che i risultati inferenziali che se ne otterrebbero utilizzando invece la (2.4), sarebbero i medesimi. Infatti coinciderebbero sia le stime di massima verosimiglianza che la matrice di informazione osservata. Tuttavia ora è lecito utilizzare la matrice

di informazione attesa senza incorrere in valori complessi, che sarebbero di difficile interpretazione ed inoltre più complicati da gestire a livello di calcolo numerico.

Nonostante le dimensioni trascurabili di questi valori, non si vede l'utilità di rocambolesche approssimazioni quando una soluzione più corretta è facilmente disponibile.

3.9 Test log-rapporto di verosimiglianza

Un metodo alternativo per condurre verifiche d'ipotesi sui coefficienti β è il test log-rapporto di verosimiglianza. La sua definizione è

$$W_p(\beta_0) = 2\big(l_p(\hat{\beta}) - l_p(\beta_0)\big). \tag{3.18}$$

Il suo utilizzo richiede la risoluzione di un problema di ottimo vincolato in quanto nella funzione $l_p(\beta_0)$ i termini di disturbo devono essere massimizzati. Per questa ragione è necessario il calcolo analitico della seguente quantità

$$\min\, l(\theta; y) \qquad \text{sotto il vincolo} \qquad H\beta = C.$$

Il teorema dei moltiplicatori di Lagrange garantisce che le soluzioni del problema di ottimo vanno ricercate tra le soluzioni del seguente sistema

$$\begin{cases} \frac{\partial}{\partial\beta} l(\theta; y) = H^T\gamma \\ H\beta = C \end{cases},$$

dove γ è il vettore dei moltiplicatori di Lagrange. Si sono omesse la stime di σ^2 e λ perchè esse coincidono con quelle usuali, sostituendovi le nuove stime di β. Come si è visto in precedenza, il risultato che se ne ottiene è

$$\begin{cases} -\frac{1}{\sigma^2}(-X^T y_\lambda + X^T X\beta) = H^T\gamma \\ H\beta = C \end{cases}.$$

Focalizzando l'attenzione sul primo termine si ottiene che

$$\begin{aligned} -\frac{1}{\sigma^2}(X^TX)^{-1}(-X^Ty_\lambda) - \frac{1}{\sigma^2}(X^TX)^{-1}X^TX\beta &= (X^TX)^{-1}H^T\gamma, \\ \beta = (X^TX)^{-1}(X^Ty_\lambda) - \sigma^2(X^TX)^{-1}H^T\gamma, \\ \beta = \hat{\beta} - \sigma^2(X^TX)^{-1}H^T\gamma. \end{aligned}$$

Sostituendo la quantità ottenuta nel vincolo si ottiene che

$$H\Big(\hat{\beta} - \sigma^2(X^TX)^{-1}H^T\gamma\Big) = C,$$
$$-\sigma^2 H(X^TX)^{-1}H^T\gamma = C - H\hat{\beta}$$

Definendo $K = \big(H(X^TX)^{-1}H^T\big)^{-1}$, si ottiene quindi che

$$\hat{\gamma} = -\frac{1}{\sigma^2}K(C - H\hat{\beta})$$

Risostituendo il termine γ nell'equazione precedente si ottiene in conclusione che

$$\hat{\beta}_0 = \hat{\beta} + (X^TX)^{-1}H^TK(C - H\hat{\beta}).$$

Le stime per σ^2 e per λ derivano di conseguenza e vengono calcolate la prima analiticamente, la seconda numericamente in maniera analoga a quanto visto finora.

Capitolo 4

Verifiche e simulazioni

4.1 Introduzione

Nel capitolo precedente sono state trattate le trasformazioni dal punto di vista teorico. Scopo di questo capitolo è invece presentare la funzione `Boxcox` per il software R, con la quale si otterranno tutte le quantità di verosimiglianza viste finora. Questa funzione verrà poi sfruttata per condurre delle simulazioni.

Una prima parte del capitolo sarà dedicata a verificare la correttezza di questa funzione, confrontandola, ove possibile, con altre librerie già presenti all'interno del software. Una seconda parte sarà volta a verificare sia la consistenza che la normalità degli stimatori, tramite simulazioni. Infine ci si occuperà di mettere in luce quali possono essere le conseguenze che derivano dall'assumere il parametro λ come noto. Inoltre, per fornire un esempio pratico, verranno ripresi i dati `cars`.

4.2 La funzione `Boxcox`

La funzione, come è riportata nel Codice A.1 a pagina 69, richiede come argomento il modello di cui si vuole effettuare la trasformazione di variabile. Il suo ouput è costituito da una lista che comprende:

- un `dataframe`, denonominato `Coefficienti`, contenente le informazioni relative ai coefficienti β: stime, deviazioni standard, test statistici;
- un `dataframe`, denonominato `lambda`, contenente le informazioni relative al parametro λ: stima, deviazione standard, test statistico per la verifica di $\lambda = 1$;
- un intervallo di confidenza per λ con livello di significatività α, denominato `Intervallo_lambda`;
- la stima di massima verosimiglianza per σ^2, denominata `Stima_varianza`;
- un oggetto `matrix` contenente la stima della matrice di informazione attesa, $I(\hat{\theta})$, denonimata `Informazione_attesa`;
- un oggetto `matrix` contenente la stima della matrice di informazione osservata, $j(\hat{\theta})$, denominata `Informazione_osservata`;
- un grafico della log-verosimiglianza profilo per λ in cui è segnalata la sua stima e un intervallo di confidenza, disattivabile ponendo come argomento `plot=FALSE`.

Tutti i test statistici e gli intervalli di confidenza si basano sul test di Wald in cui si è utilizzata la matrice di informazione attesa $I(\hat{\theta})$. All'occorrenza si può selezionare un livello di significatività diverso da quello predefinito, ovvero `alpha=0.05`, inserendolo come argomento. Un esempio di output è riportato nel Codice 4.1.

4.3 Verifica della correttezza del codice

Prima di iniziare a trarre una qualunque conclusione è necessario sincerarsi della bontà del codice utilizzato. Per fare ciò si è scelto, in alcuni casi, di confrontare i risultati prodotti dalla funzione `Boxcox` con quelli che si otterrebbero utilizzando altre funzioni. Altre volte questo non sarà possibile perchè potrebbero non esistere in R delle funzioni analoghe. In questi casi si sceglierà un approccio differente.

Codice 4.1: Esempio di output per la funzione `Boxcox` utilizzando il dataframe `cars`

```
Boxcox(lm(dist~speed))

#$Coefficienti
#                Stima  StdError    TestZ     Pvalue
#(Intercept) 1.0466220 1.0536595 0.993321 0.32055357
#speed       0.5064258 0.2203969 2.297790 0.02157372

#$lambda
#      Stima StdError_Wald     Test       Pvalue
#1 0.4305987     0.1208956 -4.70986 2.478867e-06

#$Intervallo_lambda
#[1] 0.1936476 0.6675497

#$Stima_varianza
#[1] 2.8362

#$Informazione_Attesa

#              17.62922   271.490   0.000000  -374.90744
#             271.49000  4663.987   0.000000 -6635.20490
#               0.00000     0.000   3.107894   -62.40306
#            -374.90744 -6635.205 -62.403057 10831.22406

#$Informazione_Osservata

#               17.62922   271.490   0.000000  -375.18496
#              271.49000  4663.987   0.000000 -6624.82784
#                0.00000     0.000   3.107894   -62.33529
#             -375.18496 -6624.828 -62.335291 10797.92233
```

4.3.1 Stima per λ

La stima di massima verosimiglianza, nella funzione `Boxcox`, è ottenuta tramite l'algoritmo di Newton-Raphson, seguendo il metodo illustrato nel capitolo precedente. Si deve confrontare quindi la stima per λ, calcolata con questo metodo, con quella che si ottiene tramite la funzione `nlminb`, nella quale si massimizza la log-verosimiglianza profilo. Se le due stime per λ fossero uguali, conciderebbero di conseguenza anche quelle per β e σ^2 che sono, come si è visto, calcolate esplicitamente. Per far ciò si è definita la funzione `nlogprofilo`, che è la funzione di log-verosimiglianza profilo cambiata di segno: $-l_p(\theta; y)$, presentata nel codice Codice A.2 a pagina 74.

Si è condotta una simulazione: sono stati generati 1000 differenti modelli (supponendo $\lambda = 0$), in cui $n = 50$. Per ciascuno di questi modelli si è stimato λ tramite i due differenti metodi allocando i risultati nei vettori `lambda1` e `lambda2`. Viene quindi analizzato il vettore $|\lambda_1 - \lambda_2|$, di cui si forniscono alcune statistiche descrittive. I comandi per la simulazione sono dati nel Codice A.3 a pagina 75. La Tabella 4.1 riporta i risultati ottenuti. Eccezion fatta per

Tabella 4.1: Statistiche descrittive per $|\lambda_1 - \lambda_2|$,

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.00000	0.00000	0.00000	0.00003	0.00000	0.00709

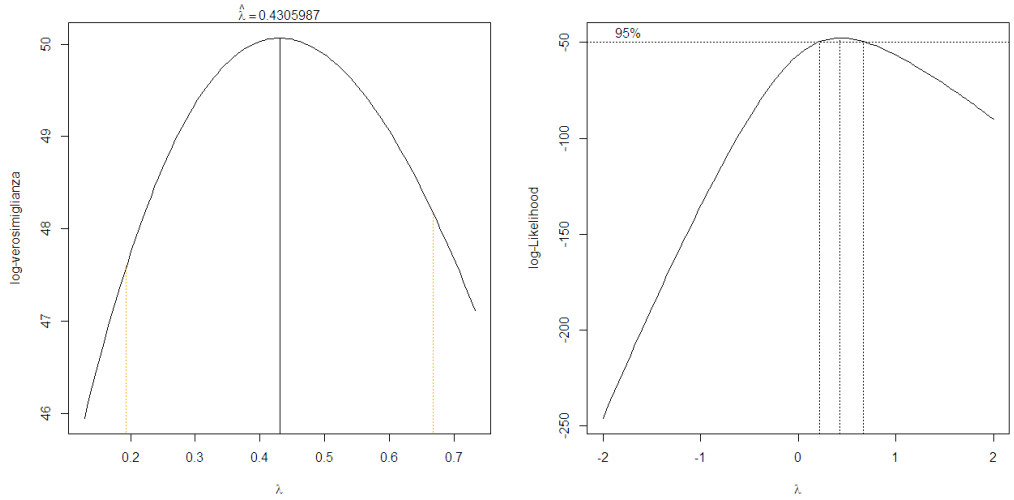
alcuni valori anomali, i due metodi portano agli stessi risultati. L'indicatore più rilevante è il terzo quartile di $|\lambda_1 - \lambda_2|$, pari esattamente a 0 che indica un errore praticamente nullo per la gran parte dei casi.

Utilizzando i dati `cars`, invece, si possono confrontare i grafici in Figura 4.1, prodotti dalla funzione `Boxcox` e dalla funzione `boxcox` della libreria `MASS`. Si tenga presente che la prima costruisce intervalli di confidenza basandosi sul test di Wald, la seconda sul test log-rapporto di verosimiglianza e perciò questi non coincidono, pur essendo molto simili.

4.3.2 Matrice di informazione osservata

Di interesse maggiore è la verifica della correttezza della versione analitica della matrice $j(\theta)$. A tale scopo, si può utilizzare il comando `hessian`

Figura 4.1: Grafico funzioni `Boxcox` e `boxcox`, utilizzando il dataframe `cars`



della libreria `numDeriv` per un confronto numerico. È necessaria, però, qualche operazione aggiuntiva perchè deve essere definita una nuova funzione, quella di verosimiglianza, che sarà poi utilizzata come argomento della funzione `hessian`. Si è deciso di definirla di segno invertito per poter ottenere automaticamente la matrice di informazione osservata. La funzione di verosimiglianza cambiata di segno, `nlog.ver`, è definita nel Codice A.4 a pagina 75. Le stime di massima verosimiglianza sono ottenute anch'esse tramite calcolo numerico. Per confrontare i risultati si è scelto di utilizzare la seguente quantità

$$\mathrm{tr}\Big((j_1(\theta) - j_2(\theta))^2\Big), \tag{4.1}$$

che rappresenta la somma di ciascun elemento della matrice differenza preso al quadrato. Vengono generati 1000 modelli, a cui viene applicata la (4.1), ed il risultato viene allocato nel vettore `quadJ`. Questo processo è eseguito nel Codice A.5 a pagina 75. Le statistiche descrittive, riportate in Tabella 4.2,

Tabella 4.2: Statistiche descrittive per il vettore `quadJ`,

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.00000	0.00000	0.00000	0.02030	0.00000	16.46000

confermano che, ad eccezione di qualche errore di approssimazione numerica, le due stime portano circa allo stesso risultato. Anche in questo caso il terzo quartile è nullo ed indica che nella gran parte dei casi le stime sono praticamente coincidenti.

Confronto matrici dei dati `cars`

Viene ora riportato un esempio, in cui si confrontano le matrici ottenute nei due metodi, utilizzando i dati `cars`. La prima si ottiene semplicemente con i comandi seguenti

```
attach(cars)
Boxcox(lm(dist~speed))$Informazione_Osservata
```

La seconda invece si ottiene tramite i comandi

```
theta.cappello<-c(1.0466220,0.5064258,2.8362,0.4305987)
hessian(function(p) nlog.ver(dist,speed,p[1],p[2],p[3], p[4]), theta.
    cappello)
```

Il risultato è riportato nella Tabella 4.3. A meno di errori trascurabili, si giunge allo stesso risultato. Con poche righe di codice si è riusciti ad ottenere un risultato equivalente, tramite le librerie già presenti. Pur essendo leggermente meno preciso, lo sforzo per il calcolo analitico potrebbe sembrare ingiustificato. Questo non è vero, in questo caso, perchè si è ottenuta anche la matrice di informazione attesa.

4.3.3 Matrice di informazione attesa

Non è possibile verificare la correttezza del codice per la matrice di informazione attesa tramite un pacchetto già presente nel software R, perchè non ne esiste uno adeguato allo scopo. Quindi si è scelto di mostrare che, utilizzando una numerosità campionaria elevata, la matrice $j(\theta)$ converge a $I(\theta)$. Pur non essendo una condizione sufficiente, è comunque un risultato confortante. Il modello dal quale si sta simulando è

$$\log(y) = x + \varepsilon, \tag{4.2}$$

Tabella 4.3: Matrice $j(\hat{\theta})$ per i dati `cars`

(a) Metodo analitico

(Intercept)	β_1	σ^2	λ
17.62922	271.49000	0.00000	-375.18496
271.49000	4663.98658	0.00000	-6624.82784
0.00000	0.00000	3.10789	-62.33529
-375.18496	-6624.82784	-62.33529	10797.92233

(b) Metodo numerico

(Intercept)	β_1	σ^2	λ
17.62922	271.49002	0.00000	-375.18505
271.49002	4663.98702	0.00008	-6624.82942
0.00000	0.00008	3.10790	-62.33543
-375.18505	-6624.82942	-62.33543	10797.92750

con $\varepsilon \sim N_n(0, I_n)$, ovvero si sta implicitamente assumendo $\lambda = 0$. Si sono generate 2000 osservazioni indipendenti ed identicamente distribuite. I comandi per questa simulazione sono

```
set.seed(11)
x<-runif(2000,0,5)
yl<-exp(x+rnorm(2000))
stima<-Boxcox(lm(yl~x))
```

I risultati sono riportati nella Tabella 4.4. Poichè le due matrici paiono abbastanza simili, pare piuttosto plausibile che i due risultati possano convergere, all'aumentare della numerosità campionaria.

4.4 Normalità degli stimatori

La teoria della verosimiglianza garantisce che gli stimatori utilizzati abbiano buone proprietà. Poichè in seguito si utilizzerà la normalità approssimata degli stimatori si vuole mostrare che questa è un'ipotesi plausibile. Anche questo viene verificato tramite una simulazione. Per n sufficientemente elevato lo stimatore di massima verosimiglianza ha distribuzione normale.

(c) Matrice di informazione attesa

(Intercept)	β_1	σ^2	λ
1854.58	4669.65	0.00	-9165.76
4669.65	15592.38	0.00	-32989.32
0.00	0.00	859.87	-4704.75
-9165.76	-32989.32	-4704.75	109022.59

(d) Matrice di informazione osservata

(Intercept)	β_1	σ^2	λ
1854.58	4669.65	0.00	-9165.35
4669.65	15592.38	0.00	-33055.40
0.00	0.00	859.87	-4704.77
-9165.35	-33055.40	-4704.77	109668.21

Tabella 4.4: Risultati della simulazione, convergenza delle matrici $I(\theta)$ e $j(\theta)$

Il processo generatore dei dati è

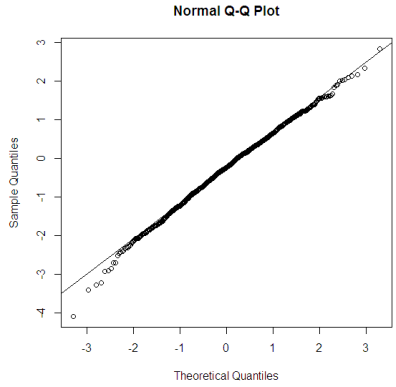
$$\frac{y^\lambda - 1}{\lambda} = 0.5x + \varepsilon, \tag{4.3}$$

con $\varepsilon \sim N_n(0, I_n)$ e $\lambda = 0.5$. La quantità pivotale di cui è nota la distribuzione è la seguente

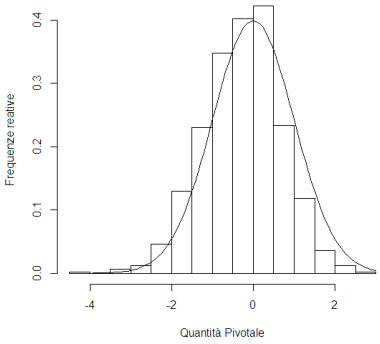
$$\frac{\hat{\beta}_1 - 0.5}{\sqrt{I(\theta)^{\beta_1\beta_1}}} \quad \dot{\sim} \quad N(0,1). \tag{4.4}$$

Il coefficiente β è stimato per $N = 1000$ volte, con $n = 200$, allocando ciascun valore nel vettore `beta.stima` mentre gli standard error vengono conservati nei vettori `SeJ` e `SeI`. Nel Codice A.7 a pagina 76 si effettua questa specifica simulazione. In Figura 4.2 vi è una diagnostica grafica per verifica della normalità della quantità pivotale oggetto d'analisi. Si applica inoltre il test di Shapiro-Wilks, il cui p-value risulta essere circa 0.06. Alla luce di ciò, si accetta l'ipotesi di normalità dello stimatore nonostante l'asimmetria manifestata dai dati. Si deve infatti tenere conto che la normalità è garantita solo asintoticamente e non per campioni finiti.

Per meglio comprendere quanto sia plausibile assumere la normalità dello stimatore, questa simulazione viene replicata cambiando alcuni parametri.



(a) Il grafico quantile contro quantile



(b) Frequenze e approssimazione gaussiana

Figura 4.2: Grafici diagnostici per la normalità dello stimatore con $N = 1000$ e $n = 200$

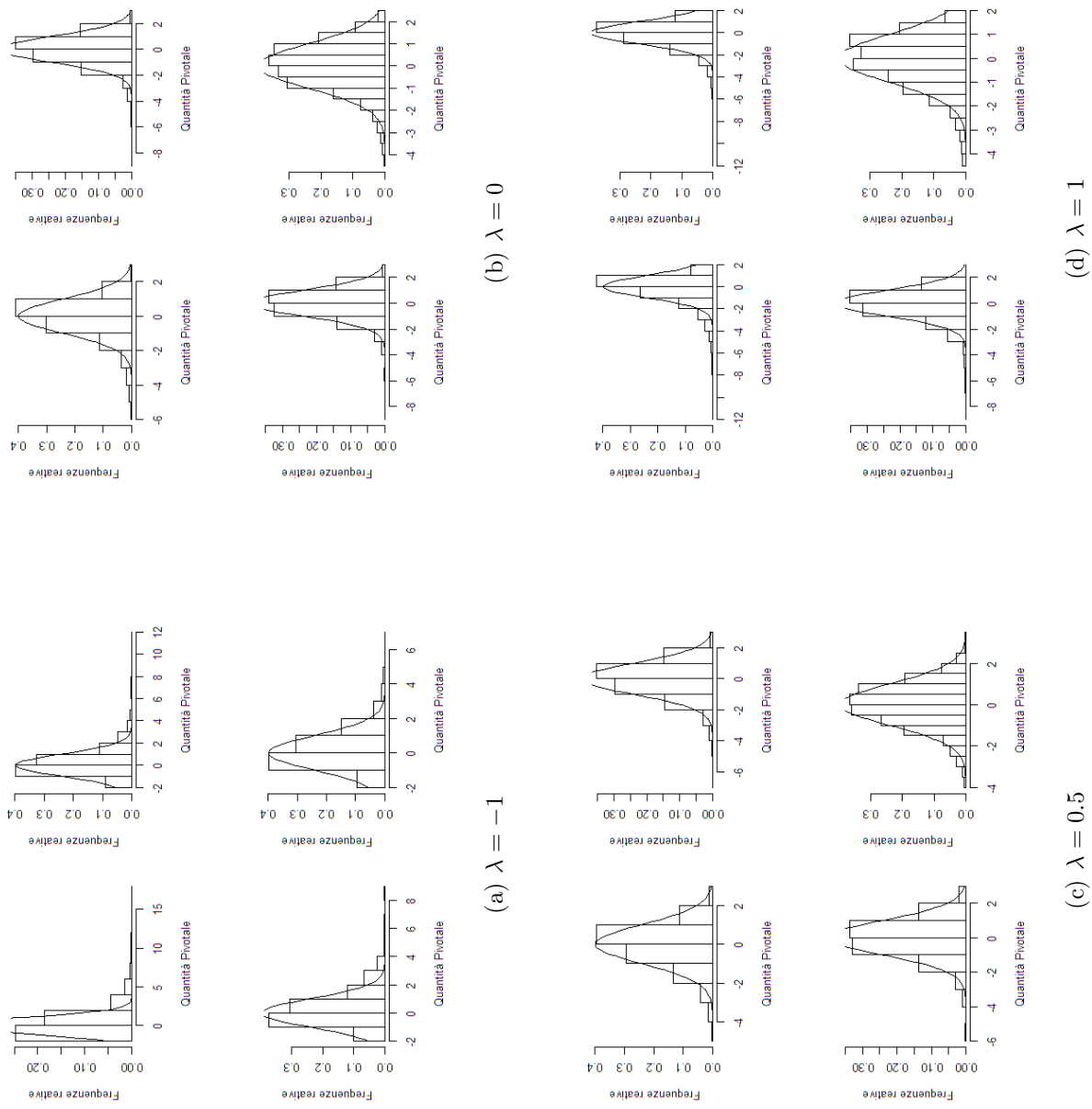
Si assumerà $n = 50, 100, 150, 200$ e $\lambda = -1, 0, 0.5, 1$. Nella Tabella 4.5 viene riportato l'errore medio per β_1 definito come

$$\text{Errore Medio} = \frac{\sum_{i=1}^{N} \hat{\beta}_{1i}}{N} - \beta_1. \tag{4.5}$$

I grafici diagnostici per queste simulazioni sono riportati in Figura 4.3. Ciò che se ne deduce è che, nella maggior parte dei casi, la convergenza alla normalità non avviene. Lo stimatore pare essere distorto anche se converge lentamente al vero valore, all'aumentare della numerosità campionaria. La velocità di convergenza dipende dal valore dei parametri e va valutata caso per caso. Ciò suggerisce una certa cautela al momento della verifica d'ipotesi soprattutto quando non si dispone di una grande quantità di dati.

	$\lambda = -1$	$\lambda = 0$	$\lambda = 0.5$	$\lambda = 1$
$n = 50$	-0.36	0.023	0.0015	0.0444
$n = 100$	-0.19	0.011	0.0002	0.0264
$n = 150$	-0.09	0.009	0.0001	0.0204
$n = 200$	-0.06	0.009	0.0001	0.0112

Tabella 4.5: Errore Medio per $\hat{\beta}_1$



(a) $\lambda = -1$ (b) $\lambda = 0$ (c) $\lambda = 0.5$ (d) $\lambda = 1$

Figura 4.3: Grafici diagnostici per la normalità di $\hat{\beta}_1$ per $n = 50, 100, 150, 200$ in senso orario a partire dalla prima in alto a sinistra

4.5 Test log-rapporto di verosimiglianza profilo per λ

Per ulteriore conferma della bontà dei test, si è implementato anche il test log-rapporto di verosimiglianza profilo per λ che necessita delle funzioni `Boxcox` e `nlogprofilo`. La funzione è data nel Codice A.6 a pagina 76. Un semplice esempio è dato dal confronto tra gli intervalli di confidenza ottenuti col test $W_p(\lambda)$ e quelli alla Wald. I risultati non dovrebbero coincidere esattamente, ma essere quantomeno simili. Nei dati `cars`, infatti, per $\alpha = 0.05$, si ha che

$$\begin{aligned} \text{Test Wald} \qquad & IC_1 = [0.1936; 0.6675], \\ \text{Test } W_p \qquad & IC_2 = [0.2203; 0.6696]. \end{aligned}$$

Si noti anche che il primo risulta essere simmetrico rispetto a $\hat{\lambda}$ per costruzione, ma ciò non si verifica anche per il secondo.

4.6 Simulazioni ed intervalli di confidenza

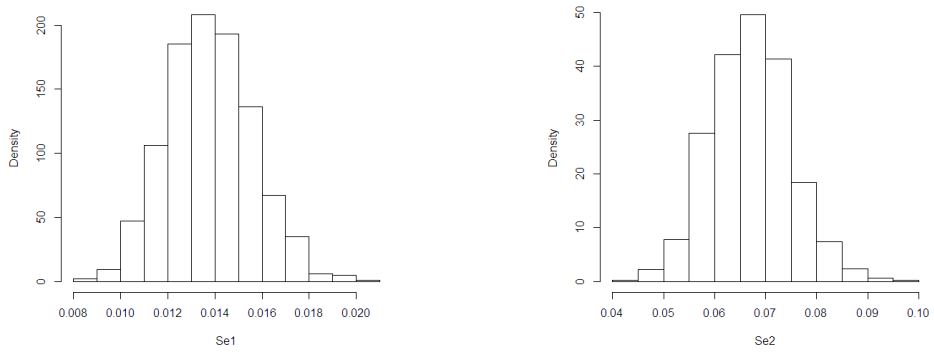
Vengono ora fatte delle simulazioni per mostrare l'entità della distorsione sulle deviazioni standard dei coefficienti β nel caso si assuma λ noto. Se la specificazione è corretta, infatti, l'intervallo di confidenza con $\alpha = 0.05$ per il coefficiente β_j dovrebbe contenere il vero valore il 95% dei casi.

Vengono simulati $N = 1000$ modelli in cui si utilizza un'estensione della variabile `speed` che assume numerosità campionaria pari a 400. Si è artificiosamente aumentato l'ampiezza del campione per permettere una migliore convergenza degli stimatori alla distribuzione normale. Si assume che i "veri valori" siano pari alle stime di massima verosimiglianza, ottenute in precedenza per i dati `cars`, e che la specificazione sia corretta. In altre parole si assume che

$$\frac{\texttt{dist}^{\hat{\lambda}} - 1}{\hat{\lambda}} = \hat{\beta}_0 + \hat{\beta}_1 \texttt{speed} + \varepsilon. \tag{4.6}$$

Nel Codice A.8 a pagina 77 sono mostrati i comandi che generano i modelli, calcolano la stima per β_1 e per la sua deviazione standard.

La differenza tra le deviazioni standard è sistematica: la variabilità aumenta mediamente di 4 volte. In Figura 4.4 è data una rappresentazione di queste quantità. Vengono ricavati gli intervalli di confidenza per entrambi gli



(a) Approccio con λ supposto noto (b) Approccio con λ ignoto

Figura 4.4: Istogramma della deviazioni standard di $\hat{\beta}_1$

approcci: supponendo λ noto o lasciandolo ignoto. Si controlla, poi, quante volte effettivamente il vero valore β_1 è contenuto nell'intervallo. I comandi utilizzati sono

```
Conf.int1<-cbind(Coeff-1*Se1*qt(0.975, n-2),Coeff+1*Se1*qt(0.975, n-2))
Conf.int2<-cbind(Coeff-1*Se2*qnorm(0.975),Coeff+1*Se2*qnorm(0.975))
Perc1<- 1-(sum(beta1<Conf.int1[,1])+sum(beta1>Conf.int1[,2]))/N
Perc2<- 1-(sum(beta1<Conf.int2[,1])+sum(beta1>Conf.int2[,2]))/N
```

Il risultato è piuttosto netto: nel caso in cui λ sia supposto noto, solamente nel 32.2% delle volte il vero valore è incluso nell'intervallo di confidenza. Nell'altro caso invece la percentuale sale al 94.2%, molto più vicina alla teorica 95%. Per $N = 1000$, ciò è da ricondursi all'errore Montecarlo: di fatto si può assumere una copertura pari al 95%. Nonostante l'ipotesi di normalità non sia pienamente soddisfatta i risultati sono coerenti con le aspettative. Osservando le statistiche descrittive della Tabella 4.6 si intuisce immediatamente come mai questo avvenga: si sottostima la variabilità dello stimatore del coefficiente β. È piuttosto confortante invece il fatto che la deviazione standard della stima dei coefficienti, `sd(Coeff)`, sia vicina alla media delle

sd(Coeff)	mean(Se2)	mean(Se1)
0.0668779	0.0670876	0.0137689

Tabella 4.6: Statistiche descrittive per la simulazione

deviazioni standard, `mean(Se2)`. Seguendo vie differenti si è giunti al medesimo risultato: `sd(Coeff)` rappresenta una stima bootstrap della deviazione standard del coefficiente. É un metodo altrettanto valido ma che comporta sforzi computazionali maggiori.

Si è già detto che nella gran parte dei casi si preferisce stimare il modello tramite un'approssimazione di λ all'intero più vicino. Verrà ricondotta la stessa simulazione aggiungendo il comando

```
lambda.stima<- round(lambda.stima*2)/2
```

e imponendo

```
Boxcox(m, plot=FALSE, lambdanoto=lambda.stima)
```

Ciò che si ottiene è una distorsione dei coefficienti, i quali non convergono più al vero valore. Si è registrato che

$$\frac{1}{N}\sum_{i=1}^{N}\hat{\beta}_{1i} = 0.644. \tag{4.7}$$

Su 1000 simulazioni il vero valore β_1 **non è mai rientrato nelle bande di confidenza**. Sostituire un'approssimazione di λ nella trasformazione, quindi, amplifica gli effetti negativi di distorsione in quanto anche la stima dei coefficienti, ora, risulta distorta. Perciò il metodo utilizzato di consueto aggrava ulteriormente la situazione: le analisi sono condotte sia trascurando parte della variabilità degli stimatori, sia con stime distorte.

4.7 Test log-rapporto di verosimiglianza

Un metodo completamente differente consiste nell'utilizzo del test $W_p(\beta)$. I risultati dovrebbero essere equivalenti a quelli ottenuti con il test di Wald nel Paragrafo 4.6. Si è condotta una simulazione simile sfruttando la funzione

`Wp.beta` riportata nel Codice A.10 a pagina 78 che consente il calcolo del test in questione e gli intervalli di confidenza. A sua volta, questa funzione necessita della `logprofilo.beta`, riportata nel Codice A.9 a pagina 78 che, come il nome suggerisce, è la log-verosimiglianza profilo per β.

Si è condotta una simulazione assolutamente analoga alla precedente allo scopo di evidenziare le eventuali differenze da quanto fatto poc'anzi. Il Codice A.11 a pagina 79 contiene i passaggi per esteso. Non sorprendentemente, i risultati portano a considerazioni molto simili: i nuovi intervalli di confidenza comprendono il vero valore di β il 94.6% delle volte. Migliore invece è la distribuzione della statistica sotto l'ipotesi nulla già utilizzando $n = 30$. Sotto H_0 infatti vale che

$$W_p(\beta) \quad \dot{\sim} \quad \chi^2_1,$$

poichè un solo parametro è stato vincolato. Nella Figura 4.5 v'è una diagnostica grafica per questa ipotesi. Complessivamente, quindi, entrambi i test concordano nel suggerire un aumento della variabilità dei coefficienti. Tuttavia il test di Wald è più facilmente interpretabile mentre il test di log-rapporto di verosimiglianza converge alla distribuzione nulla molto più rapidamente ed è quindi più affidabile per i piccoli campioni.

4.8 Rianalisi dei dati `cars`

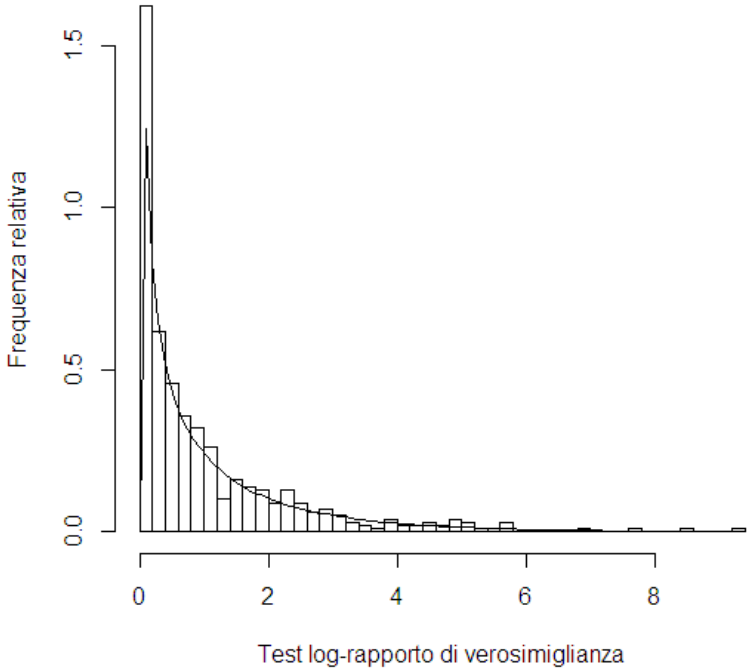
Vengono ora confrontati i due differenti approcci: quello utilizzato di consueto, in cui si assume λ noto e pari alla stima di massima verosimiglianza, e quello in cui λ è invece ignoto.

Codice 4.2: Comandi per la stima del modello, approcci differenti

```
lambda<-0.4305987
distl<-(dist^lambda-1)/lambda
m1<-lm(distl~speed)
summary(m1)
Boxcox(lm(dist~speed))
```

Le stime dei coefficienti, riportate in tabella Tabella 4.7, sono le medesime, come già fatto notare in precedenza. Le deviazioni standard invece sono invece leggermente maggiori. Ciò comporta un **allargamento degli intervalli**

Figura 4.5: Diagnostica grafica per la verifica della distribuzione χ^2_1 con $N = 500$, $n = 30$



(a) Output in cui λ è supposto noto

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	1.0466	0.7553	1.39	0.1723
speed	0.5064	0.0464	10.91	0.0000

(b) Output in λ è ignoto

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	1.0466	1.0536	0.9933	0.3205
speed	0.5064	0.2204	2.2978	0.0216

Tabella 4.7: Le stime del modello tramite i due metodi

di confidenza e soprattutto una **differente statistica test**. In generale ciò che si verifica è un aumento della variabilità delle stime. In questo caso particolare, tuttavia, le conclusioni che se ne traggono sono le medesime. Entrambi gli approcci suggeriscono di includere la variabile `speed` nel modello, anche se a differenti livelli di confidenza.

4.8.1 Intervalli di confidenza per la media e di previsione

I due approcci conducono alla stessa stima puntuale ma sono differenti, tuttavia, anche gli intervalli di confidenza per i valori medi e gli intervalli di previsione. Nei modelli lineari, si ha che

$$\hat{Y} = PY \sim N_n(X\beta, \sigma^2 P), \qquad \text{che implica che} \qquad Y_i \sim N(\mu_i, \sigma^2 h_i),$$

da cui si ricavano gli intervalli di confidenza per il valor medio, calcolati punto per punto. Il vettore h è la diagonale della matrice di proiezione ed è il vettore dei punti leva. Per gli intervalli di previsione si ha che

$$\tilde{Y} = \hat{Y} + \varepsilon \sim N_n(\hat{Y}, \sigma^2 I_n), \qquad \text{che implica che} \qquad \tilde{Y}_i \sim N(\mu_i, \sigma^2(1 + h_i)).$$

Se si suppone che il modello dipenda da un valore ignoto di λ si ha innanzitutto che

$$\hat{\beta} \quad \dot{\sim} \quad N_p\Big(\beta, I(\theta)^{\beta\beta}\Big),$$

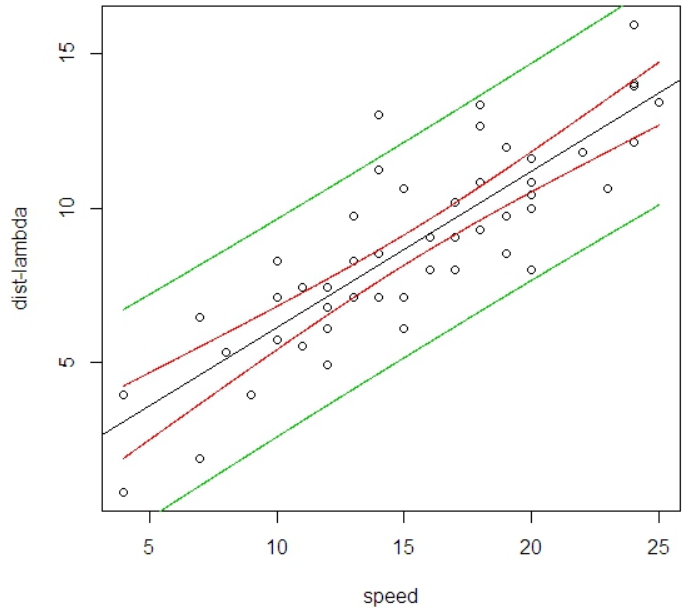
che deriva dalla teoria della verosimiglianza. Perciò poichè $\hat{Y}_\lambda = X\hat{\beta}$, vale la relazione

$$\hat{Y}_\lambda \quad \dot{\sim} \quad N_n\Big(X\beta, XI(\theta)^{\beta\beta}X^T\Big),$$

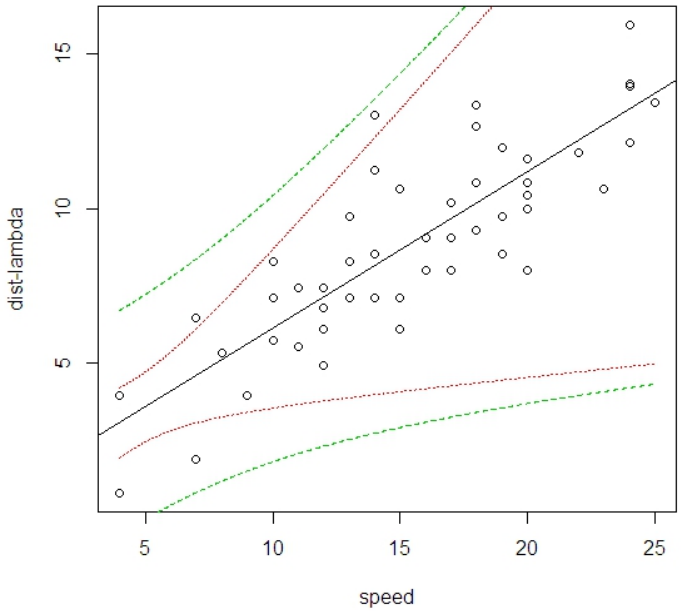
che permette di costruire intervalli di confidenza che tengano conto della variabilità di λ. Analogamente a prima, gli intervalli di previsione si costruiscono a partire da

$$\tilde{Y}_\lambda = \hat{Y} + \varepsilon \quad \dot{\sim} \quad N_n\Big(X\beta, \sigma^2 I_p + XI(\theta)^{\beta\beta}X^T\Big).$$

Sostituendo le quantità ignote con le opportune stime, si sono ottenuti intervalli di confidenza e di previsione per entrambi gli approcci, riportati in Figura 4.6. I risultati ottenuti sono completamente diversi e perciò se ne conclude che non è indifferente l'utilizzo di un metodo rispetto all'altro, come illustrato in questo esempio. Si fa notare, comunque, che gli intervalli sono stati calcolati per la variabile trasformata e non nella scala originaria.



(a) Approccio in cui λ è supposto noto



(b) Approccio in cui λ è ignoto

Figura 4.6: Intervalli di confidenza e di previsione per i dati `cars`, $\alpha = 0.05$

4.8.2 Ulteriori analisi

Nella gran parte dei casi la trasformazione non avviene scegliendo λ pari alla stima di massima verosimiglianza ma si sceglie, invece, un valore facilmente interpretabile. Nella (2.7) si è scelto infatti la radice quadrata e non la stima di massima verosimiglianza. In questi casi il calcolo della matrice di informazione osservata in un punto differente da quello di massimo può portare a risultati privi di senso come ad esempio varianze negative. La matrice di informazione attesa, invece, è definita positiva in qualunque punto essa venga calcolata. Perciò si utilizza il comando

```
Boxcox(lm(dist~speed),lambdanoto=0.5)
```

che consente di calcolare il modello in cui $\lambda = 0.5$ ma utilizzando la matrice di informazione attesa che tenga conto della sua variabilità. Il modello risultante è equivalente, a meno di trasformazioni lineari, a quello stimato nel paragrafo 2.5 a pagina 29. Il risultato, per quel che riguarda i coefficienti, è riportato in Tabella 4.8. I risultati, anche in questo caso, differiscono da

Tabella 4.8: Output modello in cui λ è arrotondato all'intero più vicino

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	0.55410	1.39444	0.39736	0.69110
speed	0.64483	0.26926	2.39483	0.01663

quelli che si otterrebbero supponendo $\lambda = 0.5$. Le deviazioni standard sono maggiori e viene nuovamente confermato il fatto che la variabilità di λ non è un dettaglio trascurabile. Non è un caso, inoltre, che la stima di β_1 coincida con quella calcolata nella (4.7). Se il modello generatore dei dati fosse quello descritto nella (4.6), si tratterebbe di una stima distorta.

Capitolo 5

Conclusioni

L'attenzione di questa relazione era posta sulla modalità con cui, normalmente, si utilizza la trasformazione Box-Cox. Trattare il parametro λ come noto è un procedimento formalmente corretto solo nel caso in cui si stia cercando una differente scala da usare come riferimento. La stima del parametro è solo un suggerimento che indica quale sia la trasformazione più opportuna.

Nella maggioranza dei casi, tuttavia, si sta effettivamente supponendo che i dati provengano da una scala ignota che dipende da λ. Perciò non tener conto del fatto che questo parametro è stato stimato comporta delle distorsioni. La dimostrazione teorica di questo fatto era già stata presentata in precedenza ed stata, in questa relazione, solamente ripresa.

Una percezione diffusa è che, se anche fossero presenti delle distorsioni, queste sarebbero irrilevanti. Tramite alcune simulazioni, confrontando gli intervalli di confidenza per le stime dei parametri, si è giunti alla conclusione che queste sono invece piuttosto consistenti. Le conseguenze possono essere anche drammatiche. Spesso inoltre si utilizza un arrotondamento di $\hat{\lambda}$ al decimale vicino e questo comporta in aggiunta una distorsione nelle stime.

Comunque si deve far notare che l'entità del 'danno' dipende dallo specifico caso in analisi e che le tecniche adottate per affrontare il problema sono difficilmente utilizzabili nella quotidianità. A volte infatti può essere utile sacrificare la correttezza formale per poter ottenere dei risultati immediatamente interpretabili e di facile comprensione. Ciò avviene, ovviamente, in contesti diversi da quello accademico.

Bibliografia

Azzalini, A. *Inferenza Statistica*. Milano: Springer Verlag, 2008.

Bickel, P.J. e A. Doksum. «An analysis of transformations revisited». In: *Journal of the American Statistical Association* 76 (1981), pp. 296–311.

Box, G.E.P. e D.R. Cox. «An analysis of transformation». In: *Journal of the Royal Statistical Society*. B (1964), pp. 211–243.

McCullagh, P. e J. A. Nelder. *Generalized linear models*. London: Chapman e Hall, 1989.

Pace, L. e A. Salvan. *Introduzione alla statistica II*. Milano: Cedam, 2001.

R Core Team. *R: A Language and Environment for Statistical Computing*. ISBN 3-900051-07-0. R Foundation for Statistical Computing. Vienna, Austria, 2012. URL: http://www.R-project.org/.

Ross, S. M. *Calcolo delle probabilita'*. Milano: Apogeo, 2007.

Sakia, R. M. «The Box-Cox Transformation Technique: A Review». In: *Journal of the Royal Statistical Society*. B (1992), pp. 169–178.

Scott, H. *A likelihood based inference on the Box-Cox family of transformation: SAS and Matlab programs*. Montana State University, 1999.

Tukey, J. W. «On the comparative anatomy of transformation diagnostics». In: *Annals of Mathematical Statistics* (1957), pp. 602–632.

Wikipedia, Modello Lineare. 2013. URL: http://it.wikipedia.org/wiki/Regressione_lineare.

Appendice A

Codice R utilizzato

Codice A.1: La funzione Boxcox

```
Boxcox<-function(model,prec=1-1e-8,plot=TRUE,alpha=0.05,eps=1e-10,
    lambdanoto=FALSE) {

        yor<-data.matrix(model$model[1])
        X<-model.matrix(model)
        n<-nrow(yor)
        I<-diag(rep(1,n))
        P2<-X%*%solve(t(X)%*%X)%*%t(X)
        p<-model$rank
        lambda<-0
        lambda2<-0
        k<-0
        #Numero massimo di iterazioni Newton Raphson
        B<-100

        ###CICLO NEWTON-RAPHSON###
        if(lambdanoto){
                lambda<-lambdanoto
                zl<- (yor^lambda-1)/lambda
                betacoef.stima<-solve(t(X)%*%X)%*%t(X)%*%as.vector(zl)
                y.stima<-as.vector(X%*%betacoef.stima)
```

```
            sigma.stima<-(t(zl)%*%(I-P2)%*%zl)/n
            u<-as.vector((lambda*(yor^lambda)*log(yor)-yor^lambda+1)/
                lambda^2)
            v<-(2-((lambda*log(yor))^2-2*lambda*log(yor)+2)*yor^lambda
                 ) /lambda^3
    }
    if(!lambdanoto){
    for(i in 1:B) {
            if(lambda==0) { zl<- log(yor) }
            if(lambda!=0) { zl<- (yor^lambda-1)/lambda }
            #Vettore u
            if(lambda==0) {u<-(log(yor)^2)/2}
            if(lambda!=0) {u<-as.vector((lambda*(yor^lambda)*log(yor)-
                yor^lambda+1)/lambda^2)}
            #Vettore v
            if(lambda==0) { v<- -(log(yor)^3)/3}
            if(lambda!=0) { v<-(2-((lambda*log(yor))^2-2*lambda*log(
                yor)+2)*yor^lambda ) /lambda^3 }
            #Sigma quadro
            sigma.stima<-(t(zl)%*%(I-P2)%*%zl)/n
            #funzione score
            g<- (-t(u)%*%(I-P2)%*%zl)/ (sigma.stima)+sum(log(yor))
            ###Varianza di lambda
            H<- ( t(v)%*%(I-P2)%*%zl-(t(u)%*%(I-P2)%*%u))/sigma.stima
                +2/n*((t(u)%*%(I-P2)%*%zl)/sigma.stima)^2
            lambda<-as.numeric(lambda-g/H)
            k<-k+1
            if(abs(lambda-lambda2)<eps) {break}
            lambda2<-as.numeric(lambda-g/H)
            }
    }
    betacoef.stima<-solve(t(X)%*%X)%*%t(X)%*%as.vector(zl)
    y.stima<-as.vector(X%*%betacoef.stima)

    #Matrice di informazione osservata
    J11<- t(X)%*%X
```

```
J12<- rep(0,p)
J13<- -t(X)%*%u
J21<- t(J12)
J22<- n/(2*as.numeric(sigma.stima) )
J23<- (-t(u)%*%(zl-X%*%betacoef.stima))/as.numeric(sigma.stima)
J31<- t(J13)
J32<- t(J23)
J33<- t(u)%*%u-t(v)%*%(zl-X%*%betacoef.stima)

J<-rbind(cbind(J11,J12,J13), cbind(J21,J22,J23), cbind(J31,J32,J33
    ) )/as.numeric(sigma.stima)
J1<-solve(J,tol=1e-150)

#Matrice di informazione attesa
#Primo ciclo
Eu<- rep(52,n)
f1<- function(Z,lambda) {
        ((Z)*log(abs(Z))-Z+1)/lambda^2
                }
f1bis<- function(Z) {
        Z*log(abs(Z))
                }
for(j in 1:n) {
                up<- qnorm(prec,1+lambda*y.stima[j],sqrt(lambda^2*
                    sigma.stima))
                down<-qnorm(1-prec,1+lambda*y.stima[j],sqrt(lambda
                    ^2*sigma.stima))
                Eu[j]   <-((integrate(function(y) f1bis(y)*dnorm(y
                    ,1+lambda*y.stima[j],sqrt(lambda^2*sigma.stima
                    )),lower=down,upper=up)$value)-lambda*y.stima[
                    j])/lambda^2
        }
#Secondo ciclo
Ezu<- rep(52,n)
f2<- function(Z,lambda){
        -(log(abs(Z)))/lambda
```

```
                }
for(j in 1:n) {
        up<- qnorm(prec,1+lambda*y.stima[j],sqrt(lambda^2*sigma.
            stima))
        down<- qnorm(1-prec,1+lambda*y.stima[j],sqrt(lambda^2*
            sigma.stima))
        Ezu[j]<- integrate(function(y) f2(y,lambda)*dnorm(y,1+
            lambda*y.stima[j],sqrt(lambda^2*sigma.stima)),lower=
            down,upper=up)$value
                }
#Terzo ciclo
D1<- rep(52,n)
D2<- rep(52,n)
f3<- function(Z,lambda,sigma.stima,y.stima)
                {
                  ((Z-(1+lambda*y.stima)) * ( 2 - Z*( log(abs(Z))
                      ^2-2*log(abs(Z))+2 ))   )/lambda^4
                }
for(j in 1:n) {
        up<- qnorm(prec,1+lambda*y.stima[j],sqrt(lambda^2*sigma.
            stima))
        down<- qnorm(1-prec,1+lambda*y.stima[j],sqrt(lambda^2*
            sigma.stima))
        D1[j]<- integrate(function(y) f1(y,lambda)^2*dnorm(y,1+
            lambda*y.stima[j],sqrt(lambda^2*sigma.stima)),lower=
            down,upper=up)$value
        }

for(j in 1:n) {
        up<- qnorm(prec,1+lambda*y.stima[j],sqrt(lambda^2*sigma.
            stima))
        down<- qnorm(1-prec,1+lambda*y.stima[j],sqrt(lambda^2*
            sigma.stima))
        D2[j]<- integrate(function(y) f3(y,lambda,sigma.stima,y.
            stima[j])*dnorm(y,1+lambda*y.stima[j],sqrt(lambda^2*
            sigma.stima)),lower=down,upper=up)$value
```

```
                }

        I11<-t(X)%*%X
        I12<-rep(0,p)
        I21<- t(I12)
        I22<- n/(2*as.numeric(sigma.stima))
        I13<- -t(X)%*%Eu
        I23<-sum(Ezu)
        I31<-t(I13)
        I32<-I23
        I33<-sum(D1)-sum(D2)

        Info<-rbind(cbind(I11,I12,I13), cbind(I21,I22,I23), cbind(I31,I32,
            I33) )/as.numeric(sigma.stima)
        I1<-solve(Info,tol=1e-150)
        std.error2<-sqrt(diag(I1))
        z.value2<-betacoef.stima/std.error2[1:p]
        p.value2<-2*(1-pnorm(abs(z.value2)))

        Tabella<-data.frame(Stima=betacoef.stima, StdError=std.error2[1:p
            ],TestZ=z.value2,Pvalue=p.value2)

        varl<-I1[p+2,p+2]
        lambdatest<-(lambda-1)/sqrt(varl)
        std.lambda<-sqrt(varl)
        p.lambda<-2*(1-pnorm(abs(lambdatest)))
        Intervallo<-lambda+c(-1,1)*qnorm(1-alpha/2)*std.lambda
        SMV<-data.frame(Stima=lambda, StdError_Wald=std.lambda , Test=
            lambdatest, Pvalue=p.lambda)

if(plot){
                logprofilo<-function(x) {
                if(x==0) { zl.profilo<- log(yor) }
                if(x!=0) { zl.profilo<- (yor^x-1)/x}
                sigma.stima.profilo<- (t(zl.profilo)%*%(I-P2)%*%zl.profilo
                    )/n
```

```
                if(sigma.stima.profilo<eps) sigma.stima.profilo=eps
                log<- -n/2*log(sigma.stima.profilo)+x*sum(log(yor))
                log}

        logprofilo<- Vectorize(logprofilo, vectorize.args = "x")
        curve(logprofilo(x),-2,2,xlab=expression("lambda"=lambda),ylab="
            log–verosimiglianza",main="Funzione di log–verosimiglianza
            profilo , intervallo Wald", sub=mtext( bquote(hat(lambda) == .(
            lambda)) ) )
        segments(x0=Intervallo[1],y0=logprofilo(-50),x1=Intervallo[1],y1=
            logprofilo(Intervallo[1]), lty="dotted",col="orange")
        segments(x0=lambda,y0=logprofilo(-50),x1=lambda,y1=logprofilo(
            lambda) )
        segments(x0=Intervallo[2],y0=logprofilo(-50),x1=Intervallo[2],y1=
            logprofilo(Intervallo[2]),lty="dotted",col="orange")
        }
list(Coefficienti=Tabella, lambda=SMV, Intervallo_lambda= Intervallo,Stima
    _varianza=as.numeric(sigma.stima),Informazione_Attesa=as.matrix(Info),
    Informazione_Osservata=as.matrix(J))
}
```

Codice A.2: Log-verosimiglianza profilo cambiata di segno

```
nlogprofilo <- function(modello,lambda) {
        X<-model.matrix(modello)
        yor<-data.matrix(modello$model[1])
        n<-nrow(yor)
        Identity<-diag(rep(1,n))
        P2<-X%*%solve(t(X)%*%X)%*%t(X)
                if(lambda==0) { zl<- log(yor) }
                if(lambda!=0) { zl<- (yor^lambda-1)/lambda}
                beta.hat<-solve(t(X)%*%X)%*%t(X)%*%as.vector(zl)
                sigma.hat<-(t(zl)%*%(Identity-P2)%*%zl)/n
        log<- -n/2*log(sigma.hat)+lambda*sum(log(yor))
-log
}
```

```
nlogprofilo<- Vectorize(nlogprofilo, vectorize.args = "lambda")
```

Codice A.3: Simulazione per la correttezza della stima di λ

```
#Allocazione vettori vuoti, definizione di n
set.seed(11)
n<-50
lambda1<-rep(0,1000)
lambda2<-rep(0,1000)
#Simulazione
for(i in 1:1000) {
        x<-runif(n)
        epsilon<-rnorm(n)
        yl<-exp(1+x+epsilon)
        m<-lm(yl~x)
lambda1[i]<-nlminb(start=1, function(par) nlogprofilo(m,par))$par
lambda2[i]<-as.numeric(Boxcox(m,plot=FALSE)$lambda[1])
}
summary(abs(lambda1-lambda2))
```

Codice A.4: Log-verosimiglianza cambiata di segno

```
nlog.ver<-function(y,x,beta0,beta1,sigma2,lambda){
           n<-length(y)
           if(lambda==0){zl<- log(y)}
           if(lambda!=0){zl<- (y^lambda-1)/lambda}
-(-n/2*log(sigma2) - 1/(2*sigma2)*sum(((zl-beta0-beta1*x)^2)) + lambda*sum
    (log(y)))
}
```

Codice A.5: Simulazione per la verifica della matrice $j(\theta)$

```
set.seed(64212)
library(numDeriv)
quadJ<-rep(0,1000)
n<-50
#Simulazione
for(i in 1:1000){
```

```
        x<-runif(n)
        epsilon<-rnorm(n)
        yl<-exp(1+x+epsilon)
        m<-lm(yl~x)
        B<-Boxcox(m,plot=FALSE)
        theta.cappello<-c(B$Coefficienti[,1],B$Stima_varianza,as.numeric(B
            $lambda[1]))
        j1<-B$Informazione_Osservata
        j2<-hessian(function(p) nlog.ver(yl,x,p[1],p[2],p[3], p[4]), theta
            .cappello)
        Diff<-j1-j2
quadJ[i]<-sum(diag(Diff%*%Diff))
}
summary(quadJ)
```

Codice A.6: Test log-rapporto di verosimiglianza profilo per λ

```
Wp<-function(modello,lambda0,alpha=0.05) {
        lambda.cappello<-as.numeric(Boxcox(modello,plot=FALSE)$lambda[1])
        Wtest<- as.numeric( 2*(-nlogprofilo(modello,lambda.cappello)+
            nlogprofilo(modello,lambda0)))
        p.value<- 1-pchisq(Wtest,1)
#Intervalli di confidenza
        int1<- uniroot(function(x) 2*(-nlogprofilo(modello,lambda.cappello
            )+nlogprofilo(modello,x))-qchisq(1-alpha,1), lower=-2, upper=
            lambda.cappello)$root
        int2<- uniroot(function(x) 2*(-nlogprofilo(modello,lambda.cappello
            )+nlogprofilo(modello,x))-qchisq(1-alpha,1), lower=lambda.
            cappello, upper=2)$root

list(Test=Wtest,PValue=p.value,Intervallo=c(int1,int2) )
}
```

Codice A.7: Esempio di simulazione per la verifica della normalità dello stimatore

```
set.seed(3456)
n<-200
x<-runif(n,0,5)
```

```
lambda<-0.5
beta.stima<-0
SeJ<-rep(0,1000)
SeI<-rep(0,1000)
for(i in 1:1000) {
        yl<-0.5*x+rnorm(n)
        y<-(1+yl*lambda)^(1/lambda)
        m<-lm(y~x)
        B<-Boxcox(m,plot=FALSE)
        beta.stima[i]<-B$Coefficienti[2,1]
        SeJ[i]<-sqrt(diag(solve(B$Informazione_Osservata)))[2]
        SeI[i]<-B$Coefficienti[2,2]
}
```

Codice A.8: Simulazione per la distorsione delle deviazioni standard

```
set.seed(11)
lambda<-0.4305987
sigma<- sqrt(2.8362)
beta0<- 1.0466220
beta1<- 0.5064258
N<-1000
Coeff<-rep(0,N)
Se1<-rep(0,N)
Se2<-rep(0,N)
x<-seq(min(speed),max(speed),length=400)
n<-length(x)
for(i in 1:N) {
        epsilon<-rnorm(n,0,sigma)
        dist.sim1<- beta0+beta1*x+epsilon
        dist.sim<-(1+dist.sim1*lambda)^(1/lambda)
        m<-lm(dist.sim~x)
        B<-Boxcox(m, plot=FALSE)
        lambda.stima<-as.numeric(B$lambda[1])
        dist.stima<-(dist.sim^lambda.stima-1)/lambda.stima
        mbis<-lm(dist.stima~x)
```

```
        Coeff[i]<-coef(mbis)[2]
        Se1[i]<-summary(mbis)$coefficients[2,2]
        Se2[i]<-B$Coefficienti[2,2]
}
```

Codice A.9: Log-verosimiglianza profilo per β

```
logprofilo.beta<- function(modello,H=t(c(0,1)),C=0) {
        minimizer<-function(modello,H,C,lambda){
                        X<-model.matrix(modello)
                        yor<-data.matrix(modello$model[1])
                        n<-nrow(yor)
                        I<-diag(rep(1,n))
                        if(lambda==0) { zl<- log(yor) }
                        if(lambda!=0) { zl<- (yor^lambda-1)/lambda}
                        K<-solve(H%*%(solve(t(X)%*%X))%*%t(H))
                        betacappello<- solve(t(X)%*%X)%*%t(X)%*%as.vector(
                            zl)
                        betacappello.0<- betacappello + solve(t(X)%*%X)%*%
                            t(H)%*%K%*%(C-H%*%betacappello)
                        sigma.cappello.0<-t((zl-X%*%betacappello.0))%*%(zl
                            -X%*%betacappello.0)/n
                log<- -n/2*log(sigma.cappello.0)+lambda*sum(log(yor))
                -log
        }
-nlminb(start=1, function(p) minimizer(modello, H,C,p))$objective
}
logprofilo.beta<-Vectorize(logprofilo.beta, vectorize.args="C")
```

Codice A.10: Test log-rapporto di verosimiglianza e intervalli di confidenza per β

```
Wp.beta<-function(modello, H=t(c(0,1)),beta, alpha=0.05){
        B<-Boxcox(modello, plot=FALSE)
        betacappello<-as.numeric(B$Coefficienti[2,1])
        down<-betacappello-10*B$Coefficienti[2,2]
        up<-betacappello+10*B$Coefficienti[2,2]
        ltheta.cappello<-logprofilo.beta(modello,H,C=betacappello)
        ltheta0<-logprofilo.beta(modello,H,C=beta)
```

```
        Wptest<- 2*(ltheta.cappello-ltheta0)
        pvalue<-1-pchisq(Wptest,1)
        int1<-uniroot(function(x) (2*(ltheta.cappello - logprofilo.beta(
            modello,H,x))-qchisq(1-alpha,1)), lower=down, upper=
            betacappello)$root
        int2<-uniroot(function(x) (2*(ltheta.cappello - logprofilo.beta(
            modello,H,x))-qchisq(1-alpha,1)), lower=betacappello, upper=up
            )$root

list(Test=Wptest, Pvalue=pvalue,Intervallo=c(int1, int2))
}
```

Codice A.11: Simulazione intervalli di confidenza con $W_p(\beta)$

```
set.seed(654)
lambda<-0.4305987
sigma<- sqrt(2.8362)
beta0<- 1.0466220
beta1<- 0.5064258
N<-500
x<-seq(min(speed),max(speed),length=30)
n<-length(x)
test<-rep(0,N)
Conf1<-rep(0,N)
Conf2<-rep(0,N)

for(i in 1:N) {
        epsilon<-rnorm(n,0,sigma)
        dist.sim1<- beta0+beta1*x+epsilon
        dist.sim<-(1+dist.sim1*lambda)^(1/lambda)
        m<-lm(dist.sim~x)
        W<-Wp.beta(m,H=t(c(0,1)),beta=beta1)
        test[i]<-W$Test
        Conf1[i]<-W$Intervallo[1]
        Conf2[i]<-W$Intervallo[2]
}
```